Chapter 6
Case Studies of C_1^k-Subdivision Algorithms

In this chapter, we formally introduce and scrutinize three of the most popular subdivision algorithms, namely the *Catmull–Clark algorithm* [CC78], the *Doo–Sabin algorithm* [DS78], and *Simplest subdivision*[1] [PR97]. Besides the algorithms in their original form, it is instructive to consider certain variants. We selectively modify a subset of weights to obtain a variety of algorithms that is rich enough to illustrate the relevance of the theory developed so far. In particular, we show that a double subdominant eigenvalue is neither necessary nor sufficient for a C_1^k-algorithm: First, there are variants of the Doo–Sabin algorithm with a double subdominant eigenvalue, which provably fail to be C_1^1 because the Jacobian determinant ${}^{\times}D\psi$ of the characteristic ring changes sign. Second, for valence $n = 3$, Simplest subdivision reveals an eightfold subdominant eigenvalue, but due to the appropriate structure of Jordan blocks, it is still C_1^1. In all cases, the algorithms are symmetric so that the conditions of Theorem 5.24/105 can be used for the analysis.

6.1 Catmull–Clark Algorithm and Variants

The Catmull–Clark algorithm (Fig. 6.1/110) is currently the most popular subdivision algorithm due to its close relationship with the tensor-product spline standard. The algorithm generalizes uniform knot insertion for bicubic tensor-product B-splines. Since each n-gon of the original mesh of control points is subdivided into n quadrilaterals the mesh is purely quadrilateral after the first step. Figure 6.2/110 defines the rules of the subdivision algorithm in terms of stencils. A *stencil* is an intuitive representation of a row of the local subdivision matrix A. In the regular case, when $n = 4$, the control points have the structure of a regular planar grid. In analogy, near an extraordinary vertex, the control points can be arranged with n-fold symmetry.

[1] In the literature, Simplest subdivision is sometimes also called *Mid-edge subdivision*.

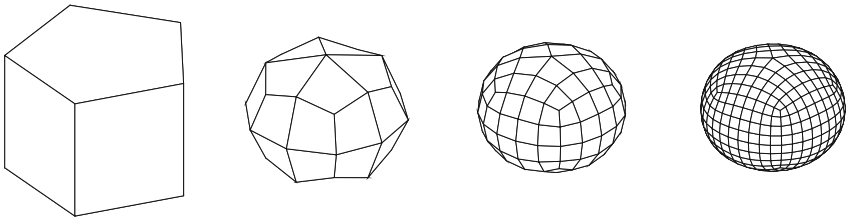

Fig. 6.1 Illustration of Catmull–Clark algorithm: Mesh refinement.

There are three types of stencils for Catmull–Clark subdivision that are inherited from bi-cubic B-spline subdivision and one generalization (see Fig. 6.2/110, *right*) that is expressed in terms of the variables

$$\alpha\,,\ \beta\,,\ \gamma,\quad \alpha+\beta+\gamma=1. \tag{6.1}$$

In [CC78], Catmull and Clark suggest

$$\alpha=1-\frac{7}{4n},\ \beta=\frac{3}{2n},\ \gamma=\frac{1}{4n}. \tag{6.2}$$

For $n = 4$, this choice coincides with the regular stencil. To establish C_1^2-smoothness for variables α,β,γ summing to 1, we first define an appropriate data structure for the space of rings. Then we determine the characteristic ring $\boldsymbol{\psi}$ and apply Theorem 5.24/105 to obtain necessary and sufficient conditions for smoothness. Let us start with considering a single ring $\mathbf{x}^m$. Each of the n segments $\mathbf{x}_j^m, j \in \mathbb{Z}_n$, consists of three bicubic B-spline patches. The corresponding vector $\mathbf{Q} = [\mathbf{Q}_0; \ldots; \mathbf{Q}_{n-1}]$ of initial data is split into n blocks $\mathbf{Q}_j$ with 13 elements each. We label the coefficients of each block $\mathbf{Q}_j = [\mathbf{q}_{j,1}; \ldots; \mathbf{q}_{j,13}]$ as shown in

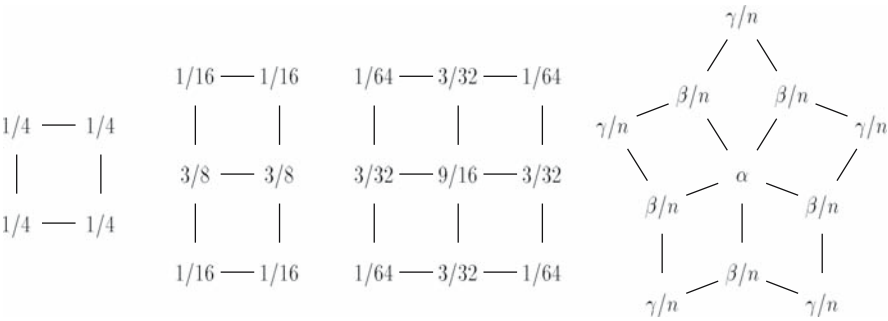

Fig. 6.2 Stencils for the Catmull–Clark algorithm: From *left* to *right*, the weights for generating a new 'face point', a new 'edge point', a new ordinary 'vertex point', and a new extraordinary 'vertex point' of valence n. The scalars α, β and γ are constrained by (6.1/110) and their originally published choice is given in (6.2/110).

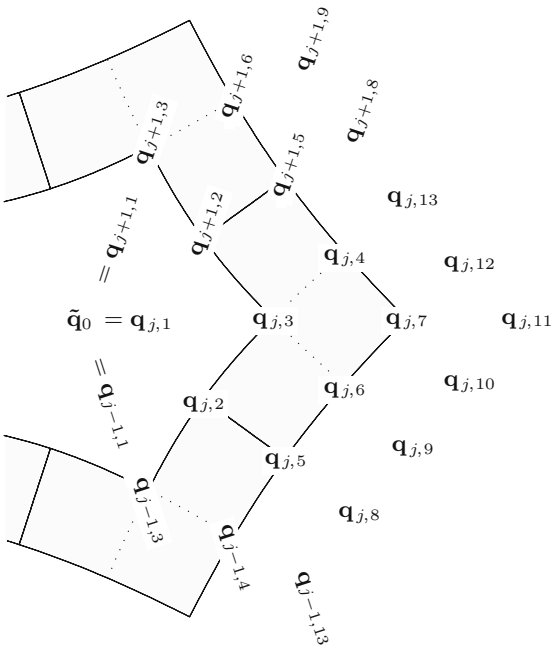

Fig. 6.3 Labelling of the control points $\mathbf{q}_{j,k}$ of the Catmull–Clark algorithm: Following (6.3/111), the center control point is replicated.

Fig. 6.3/111. Following Example 5.14/97, identical copies of the central coefficient $\tilde{\mathbf{q}}_0$ are placed in all blocks to obtain a circulant structure,

$$\tilde{\mathbf{q}}_0 = \mathbf{q}_{0,1} = \cdots = \mathbf{q}_{n-1,1}. \tag{6.3}$$

The corresponding subdivision matrix is block-circulant,

$$A = \operatorname{circ}(A_0, \ldots, A_{n-1}),$$

where the blocks A_j are (13×13)-matrices. Moreover, the algorithm is symmetric in the sense of Definition 5.21/104, and the generated segments satisfy the conditions (4.7/62) and (4.8/62) for $k = 2$. According to (5.14/99), the DFT $\hat{A} = \operatorname{diag}(\hat{A}_0, \ldots, \hat{A}_{n-1})$ of A is block-diagonal. Omitting the details, we find the following: the blocks $\hat{A}_i$ have the form

$$\hat{A}_i = \begin{bmatrix} \hat{A}_i^{0,0} & 0 & 0 \\ \hat{A}_i^{1,0} & \hat{A}_i^{1,1} & 0 \\ \hat{A}_i^{2,0} & \hat{A}_i^{2,1} & 0 \end{bmatrix}. \tag{6.4}$$

Recalling (5.12/98), we set

$$c_n + \mathbf{i}s_n := w_n := \exp(2\pi\mathbf{i}/n), \quad c_{n,i} + \mathbf{i}s_{n,i} := w_n^i = \exp(2\pi i\mathbf{i}/n),$$

for $i \in \mathbb{Z}_n$. With

$$p_1 := 1/64,\ p_2 := 3/32,\ p_3 := 9/16,\ q_1 := 1/16,\ q_2 := 3/8,\ r := 1/4,$$

and using the abbreviation $w := w_n^i$, we obtain

$$\hat{A}_i^{0,0} := \begin{bmatrix} \alpha\delta_{i,0} & \beta\delta_{i,0} & \gamma\delta_{i,0} \\ q_2\delta_{i,0} & 2q_1c_{n,i} + q_2 & q_1(1+\overline{w}) \\ r\delta_{i,0} & r(1+w) & r \end{bmatrix} \tag{6.5}$$

and

$$\begin{bmatrix} \hat{A}_i^{1,0} & \hat{A}_i^{1,1} \\ \hat{A}_i^{2,0} & \hat{A}_i^{2,1} \end{bmatrix} := \left[\begin{array}{ccc|ccc|c} q_2\delta_{i,0} & q_1 + q_2w & q_1 & q_1 & q_1w & 0 & 0 \\ p_2\delta_{i,0} & 2p_1c_{n,i} + p_3 & p_2(1+\overline{w}) & p_1\overline{w} & p_2 & p_1 & 0 \\ q_1\delta_{i,0} & q_1w + q_2 & q_2 & 0 & q_1 & q_1 & 0 \\ p_1\delta_{i,0} & p_2(1+w) & p_3 & p_2 & p_1(1+w) & p_2 & p_1 \\ \hline 0 & q_2 & q_1(1+\overline{w}) & q_1\overline{w} & q_2 & q_1 & 0 \\ 0 & r & r & 0 & r & r & 0 \\ 0 & q_1 & q_2 & q_1 & q_1 & q_2 & q_1 \\ 0 & 0 & r & r & 0 & r & r \\ 0 & q_1w & q_2 & q_2 & q_1w & q_1 & q_1 \\ 0 & rw & r & r & rw & 0 & 0 \end{array}\right].$$

The eigenvalues $1/8, 1/16, 1/32, 1/64$ of the sub-matrix $\hat{A}_i^{1,1}$ are n-fold eigenvalues of A. Other non-zero eigenvalues come only from $\hat{A}_i^{0,0}$. For $i = 0$, we obtain the obligatory eigenvalue $\lambda_0 = 1$ and, with $\gamma := 1 - \alpha - \beta$, the pair

$$\lambda_{1,2}^0 := \left(4\alpha - 1 \pm \sqrt{(4\alpha-1)^2 + 8\beta - 4}\right)/8.$$

Depending on the sign of the discriminant, these two eigenvalues are either both real or complex conjugate. For $i \neq 0$, the non-zero eigenvalues of $\hat{A}_i^{0,0}$ are always real and given by

$$\lambda_{1,2}^i := \left(c_{n,i} + 5 \pm \sqrt{(c_{n,i}+9)(c_{n,i}+1)}\right)/16.$$

Here and in the penultimate display, the subscript 1 refers to the plus sign, and the subscript 2 refers to the minus sign. By Theorem 5.18/101, the subdominant eigenvalue λ must come from the blocks $\hat{A}_1, \hat{A}_{n-1}$. Because the eigenvalue $1/32$ has algebraic multiplicity n, the only candidate is

$$\lambda := \lambda_1^1 = \lambda_1^{n-1} = \left(c_n + 5 + \sqrt{(c_n+9)(c_n+1)}\right)/16. \tag{6.6}$$

Straightforward calculus shows that

$$1 > \lambda > 1/4 > \lambda_2^i > 1/8, \quad i = 1, \ldots, n-1 \tag{6.7}$$

$$\lambda > \lambda_1^i > 1/4, \quad i = 2, \ldots, n-2\,. \tag{6.8}$$

That is, λ is subdominant if α, β, γ are chosen such that

$$\lambda > \max\{|\lambda_1^0|, |\lambda_2^0|\}. \tag{6.9}$$

We will comment on the set of feasible weights at the end of this section, but state already now that the original weights of Catmull–Clark (6.2/110) satisfy the condition.

For computing the characteristic ring, the eigenvector $\hat{v}$ of $\hat{A}_1$ is partitioned into three blocks, $\hat{v} = [\hat{v}_0; \hat{v}_1; \hat{v}_2]$, according to the structure of $\hat{A}_1$ defined in (6.4/111). Then $\hat{A}_1\hat{v} = \lambda\hat{v}$ is equivalent to

$$\begin{aligned} (\hat{A}_1^{0,0} - \lambda)\hat{v}_0 &= 0 \\ (\hat{A}_1^{1,1} - \lambda)\hat{v}_1 &= -\hat{A}_1^{1,0}\hat{v}_0 \\ \hat{v}_2 &= (\hat{A}_1^{2,0}\hat{v}_0 + \hat{A}_1^{2,1}\hat{v}_1)/\lambda. \end{aligned} \tag{6.10}$$

Now, $\hat{v}$ can be computed conveniently starting from

$$\hat{v}_0 := [1 + \overline{w}_n, 16\lambda - 2c_n - 6], \tag{6.11}$$

which solves the first eigenvector equation.

By (6.6/112), the characteristic ring depends only on $c_n \in [-1/2, 1)$ and not on the particular choice of weights α, β, γ. For the interval of definition, we can invert the relation to obtain

$$c_n = \frac{16\lambda^2 - 10\lambda + 1}{2\lambda}, \quad \lambda \in \Lambda := \big[(9 + \sqrt{17})/32,\ (3 + \sqrt{5})/8\big), \tag{6.12}$$

and write the characteristic ring in terms of $\lambda \in \Lambda$. After scaling, the eigenvector $\hat{v}$ has the form

$$\hat{v} = ((4\lambda - 1)\hat{v}_{\mathrm{re}} + 2s\lambda(64\lambda - 1)\mathbf{i}\hat{v}_{\mathrm{im}})/13020,$$

where

$$\hat{v}_{\mathrm{re}} := \begin{bmatrix} 0 \\ 4\lambda^2(64\lambda{-}1)(32\lambda{-}1)(16\lambda{-}1)(4\lambda{-}1) \\ 8\lambda^2(64\lambda{-}1)(32\lambda{-}1)(16\lambda{-}1) \\ 4\lambda^2(64\lambda{-}1)(928\lambda^2{+}228\lambda{-}31) \\ 8\lambda^2(64\lambda{-}1)(16\lambda{-}1)(4\lambda{-}1)(4\lambda{+}13) \\ 4\lambda^2(64\lambda{-}1)(928\lambda^2{+}228\lambda{-}31) \\ 80\lambda^2(1280\lambda^3{+}2128\lambda^2{-}56\lambda{-}13) \\ (64\lambda{-}1)(16\lambda{-}1)(4\lambda{-}1)(100\lambda^2{+}42\lambda{-}1) \\ 4\lambda(64\lambda{-}1)(640\lambda^3{+}688\lambda^2{-}82\lambda{-}1) \\ 20\lambda(2048\lambda^4{+}11040\lambda^3{+}812\lambda^2{-}165\lambda{-}1) \\ 40\lambda(5248\lambda^3{+}1568\lambda^2{-}133\lambda{-}5) \\ 20\lambda(2048\lambda^4{+}11040\lambda^3{+}812\lambda^2{-}165\lambda{-}1) \\ 4\lambda(64\lambda{-}1)(640\lambda^3{+}688\lambda^2{-}82\lambda{-}1) \end{bmatrix}, \quad \hat{v}_{\mathrm{im}} := \begin{bmatrix} 0 \\ -4\lambda^2(32\lambda{-}1)(16\lambda{-}1) \\ 0 \\ 140\lambda^2(8\lambda{-}1) \\ -8\lambda^2(16\lambda{-}1)(4\lambda{+}13) \\ -140\lambda^2(8\lambda{-}1) \\ 0 \\ -(16\lambda{-}1)(100\lambda^2{+}42\lambda{-}1) \\ -4\lambda(160\lambda^2{+}132\lambda{-}1) \\ -20\lambda(8\lambda^2{+}15\lambda{+}1) \\ 0 \\ 20\lambda(8\lambda^2{+}15\lambda{+}1) \\ 4\lambda(160\lambda^2{+}132\lambda{-}1) \end{bmatrix}.$$

Now, we compute $f(1, 1, 0)$ to ensure normalization. After reformatting the middle patch according to its tensor product structure and substituting in the parameter

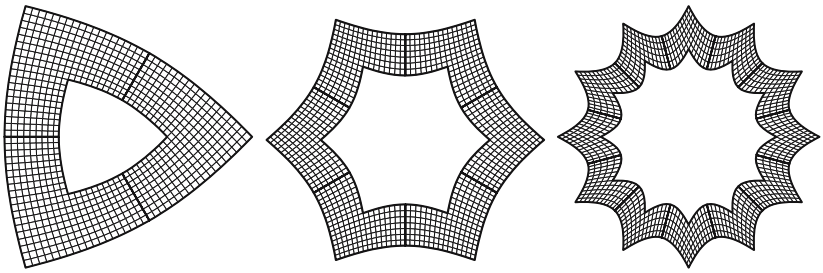

Fig. 6.4 Characteristic ring of the Catmull–Clark algorithm: (*left*) $n = 3$, (*middle*) $n = 6$, and (*right*) $n = 12$.

$\boldsymbol{\sigma} = [1, 1]$, we obtain with $b := [1/6, 2/3, 1/6]$

$$f(1,1,0) = b \begin{bmatrix} \hat{v}^3 & \hat{v}^4 & \hat{v}^{13} \\ \hat{v}^6 & \hat{v}^7 & \hat{v}^{12} \\ \hat{v}^9 & \hat{v}^{10} & \hat{v}^{13} \end{bmatrix} \cdot b = \frac{2}{29295}\lambda(4\lambda - 1)\big(139264\lambda^4 + 170496\lambda^3 + 112\,\lambda^2 - 1,476\lambda - 11\big).$$

For $\lambda \in \Lambda$ this value is real and positive. That is, the characteristic ring f is normalized in the sense of Definition 5.19/103.

To establish smoothness, we verify the sufficient conditions (5.22/107) given in Theorem 5.25/107. The derivative of f_0 in t-direction is computed by differencing the Bernstein–Bézier control points of the three bicubic patches. The elements of the resulting three sets of 3×4 coefficients are enumerated $k_1, \ldots, k_{36}$. All k_μ are polynomials in λ with rational coefficients. More precisely,

$$k_\mu(\lambda) = p_\mu(\lambda) + \mathbf{i}s_n q_\mu(\lambda), \quad \mu = 1, \ldots, 36,$$

for certain polynomials p_μ and q_μ of degree ≤ 7 in λ which *are independent* of n or the special weights. Computing the Sturm sequences of all these polynomials on the larger, but more convenient interval $\Lambda' := [0.41,\ 0.66] \supset \Lambda$, we find either no root or the single root $(3 + \sqrt{5})/8 \notin \Lambda$. Hence, the sign of all polynomials in question is constant and can be determined by evaluation at a single point. At $\lambda = 1/2$, we obtain $p_\mu(\lambda) = q_\mu(\lambda) = 3255/13020 = 1/4$ for all $\mu = 1, \ldots, 36$. Hence, all coefficients k_μ are positive so that, by the convex hull property, $\mathrm{Re}(D_2 f_0) > 0$ and $\mathrm{Im}(D_2 f_0) > 0$. Hence, by Theorem 5.25/107, the algorithm is C_1^2.

Figure 6.4/114 shows the characteristic rings for different values of n. As already mentioned above, it depends only on n, but not on the particular choice of the weights α, β, γ, provided that the conditions, summarized in the following theorem, are satisfied.

Theorem 6.1 (C_1^2-variants of Catmull–Clark). *For $n \geq 3$, $c_n := \cos(2\pi/n)$, and*

$$\lambda := \Big(c_n + 5 + \sqrt{(c_n + 9)(c_n + 1)}\Big)/16 \tag{6.13}$$

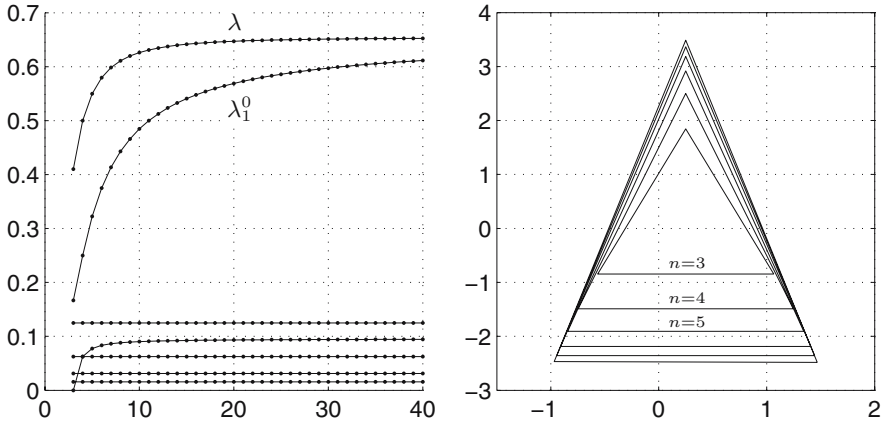

Fig. 6.5 Illustration of Theorem 6.1/114: (*left*) Spectrum of the subdivision matrix of the Catmull–Clark algorithm with standard weights (6.2/110) for $n = 3, \ldots, 40$. The subdominant eigenvalue λ satisfies the condition of the theorem. (*right*) Triangles $\Delta_3, \ldots, \Delta_8$ in the $\alpha\beta$-plane. Choosing (α, β) inside these triangles yields a C_1^2-algorithm.

$$\lambda_{1,2}^0 := \left(4\alpha - 1 \pm \sqrt{(4\alpha-1)^2 + 8\beta - 4}\right)/8, \tag{6.14}$$

the Catmull–Clark algorithm with weights α, β and $\gamma = 1 - \alpha - \beta$ is a standard C_1^2-algorithm if and only if $\lambda > \max\{|\lambda_1^0|, |\lambda_2^0|\}$.

Let us briefly comment on the set of parameters yielding a C_1^2-algorithm. We define

$$\tilde{\alpha} := \frac{\alpha}{2} - \frac{1}{8}, \quad \tilde{\beta} := \frac{\beta}{8} - \frac{1}{16},$$

and obtain the equivalent condition

$$\left|\tilde{\alpha} \pm \sqrt{\tilde{\alpha}^2 + \tilde{\beta}}\right| < \lambda.$$

Distinguishing the cases $\tilde{\alpha}^2 + \tilde{\beta} \geq 0$ and $\tilde{\alpha}^2 + \tilde{\beta} \leq 0$, we find

$$-\lambda^2 < \tilde{\beta} < \lambda(\lambda - 2|\tilde{\alpha}|\lambda).$$

Given the valence n and the corresponding subdominant eigenvalue λ, the set of pairs $(\tilde{\alpha}, \tilde{\beta})$ satisfying this condition forms the interior of a triangle $\tilde{\Delta}_n$. Accordingly, in terms of the original parameters α, β, we obtain a triangle Δ_n in the $\alpha\beta$-plane with corners

$$\big(1/4 \pm 2\lambda, 1/2 - \lambda^2\big), \quad \big(1/4, 1/2 + \lambda^2\big).$$

On the right hand side, Fig. 6.5/115 shows the triangles $\Delta_3, \ldots, \Delta_8$. On the left hand side, we see the complete spectrum of the algorithm when using the standard

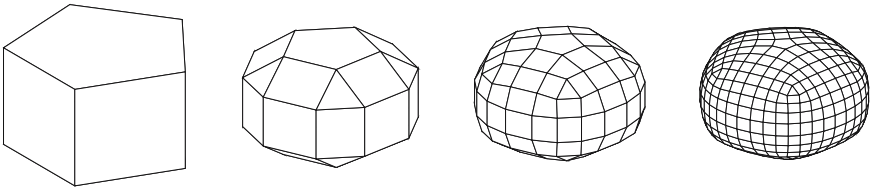

Fig. 6.6 Illustration of Doo–Sabin algorithm: Mesh refinement.

weights (6.2/110). We see that the condition of Theorem 6.1/114 is satisfied for $n = 3, \ldots, 40$, and one can show that this is also true for *all* $n > 40$. However, it should be noted that both λ and λ_1^0 are converging to the limit $(3 + \sqrt{5})/8 \approx 0.6545$ as $n \to \infty$. As we will explain in the next chapter, shape may be poor if the ratio of the subdominant eigenvalue λ and the next smaller *subsubdominant eigenvalue* is close to 1.

6.2 Doo–Sabin Algorithm and Variants

The Doo–Sabin algorithm generalizes subdivision of uniform biquadratic tensor-product B-splines. For each n-gon of the original mesh of control points, a new, smaller n-gon is created and connected with its neighbors as depicted in Fig. 6.6/116. Figure 6.7/117 shows the *stencils* for generating a new n-gon from an old one, both for the regular case $n = 4$ (*left*) and the general case (*middle*). For $n = 4$ the weights are those of the biquadratic spline. Doo and Sabin in [DS78] suggested

$$a_j := \frac{\delta_{j,0}}{4} + \frac{3 + 2\cos(2\pi j/n)}{4n} \tag{6.15}$$

for the general case. In the following, we analyze all algorithms that are affine invariant and symmetric:

$$\sum_{j=0}^{n-1} a_j = 1, \quad a_j = a_{n-j}, \quad j \in \mathbb{Z}_n. \tag{6.16}$$

Each of the n segments $\mathbf{x}_j^m, j \in \mathbb{Z}_n$, of the m-th ring generated by the Doo–Sabin algorithm consists of three biquadratic B-spline patches. Accordingly, we can split the control points $\mathbf{Q}^m$ into n groups of nine control points, each, ordered as shown in Fig. 6.7/117 (*right*).

Since the algorithm is symmetric, we can apply DFT as introduced in Sect. 5.4/95 to obtain the block-diagonal form $\hat{A} = \operatorname{diag}(\hat{A}_0, \ldots, \hat{A}_{n-1})$ of the subdivision matrix. The non-zeros elements of $\hat{A}_i$ are situated in the first four columns. With

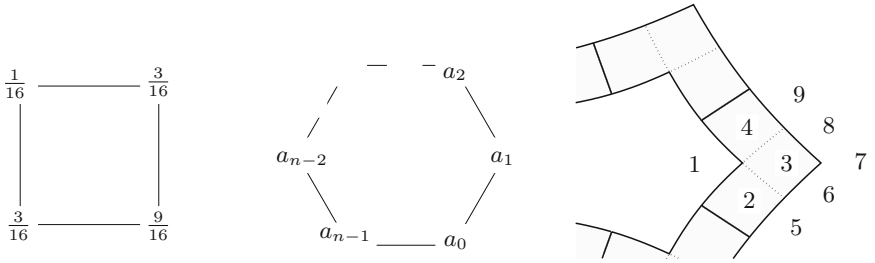

Fig. 6.7 Stencils for the Doo–Sabin algorithm: (*left*) Regular refinement rule, (*middle*) general refinement rule, and (*right*) control point labels of one segment.

$w = w_n^i = \exp(2\pi i \mathbf{i}/n)$, as before, we have

$$\hat{A}_i(:, 1:4) = \begin{bmatrix} \hat{\alpha}_i & 0 & 0 & 0 \\ p + \overline{w}q & q & 0 & \overline{w}r \\ p & q & r & q \\ p + wq & wr & 0 & q \\ q + \overline{w}r & p & 0 & \overline{w}q \\ q & p & q & r \\ r & q & p & q \\ q & r & q & p \\ q + wr & wq & 0 & p \end{bmatrix}, \tag{6.17}$$

where $p := 9/16, q := 3/16, r := 1/16$ are the standard weights for quadrilaterals, and

$$\hat{a}_i := \sum_{j=0}^{n-1} w_n^{-ij} a_j$$

are the entries of the DFT of the vector $[a_0, \ldots, a_{n-1}]$ of special weights for the inner n-gon. The weights a_j sum to one, i.e., $\hat{a}_0 = 1$, and satisfy $a_j = a_{n-j}$. Hence, $\hat{a}_i = \hat{a}_{n-i}$ is real. The eigenvalues of $\hat{A}_i$ are $\hat{a}_i$, $1/4$, $1/8$, $1/16$, 0. Since each eigenvalue $1/4$ corresponds to a separate eigenspace and also the eigenspace of each $\hat{a}_i$ is spanned by a single vector, and by the requirement on the Fourier index to be $\mathcal{F}(\lambda) = \{1, n-1\}$, the subdominant eigenvalue must be $\lambda := \hat{a}_1 = \hat{a}_{n-1} \in (1/4, 1)$ to generate a C_1^1-algorithm. This yields the inequality

$$1 > \hat{a}_1 > \max\{1/4, |\hat{a}_2|, \ldots, |\hat{a}_{n/2}|\}. \tag{6.18}$$

We will see below that this constraint is however *not sufficient*. Using a computer algebra system, one can determine the complex eigenvector $\hat{v}$ of $\hat{A}_1$ corresponding to λ explicitly:

$$\hat{v} = \begin{bmatrix} 2\lambda(16\lambda-1)(8\lambda-1)(4\lambda-1) \\ 6\lambda(16\lambda-1)(6\lambda-1+2\overline{w}_n\lambda) \\ 18\lambda(32\lambda^2-1+4c_n\lambda) \\ 6\lambda(16\lambda-1)(6\lambda-1+2w_n\lambda) \\ (16\lambda-1)\big(12\lambda^2+18\lambda-3+\overline{w}_n(4\lambda^2+12\lambda-1)\big) \\ 6\lambda\big(32\lambda^2+64\lambda-12+c_n(20\lambda+1)-is_n(16\lambda-1)\big) \\ 64\lambda^3+512\lambda^2-46\lambda-8+36c_n\lambda(2\lambda+1) \\ 6\lambda\big(32\lambda^2+64\lambda-12+c_n(20\lambda+1)+is_n(16\lambda-1)\big) \\ (16\lambda-1)\big(12\lambda^2+18\lambda-3+w_n(4\lambda^2+12\lambda-1)\big) \end{bmatrix},$$

where as before $w_n = c_n + \mathbf{i}s_n$.

In particular, for the original Doo–Sabin weights in (6.15/116), we have $\lambda = 1/2$ and, rearranging the entries of $\hat{v}$ in a (3×3)-matrix according to Fig. 6.7/117, *right*,

$$\begin{bmatrix} \hat{v}_5 & \hat{v}_6 & \hat{v}_7 \\ \hat{v}_2 & \hat{v}_3 & \hat{v}_8 \\ \hat{v}_1 & \hat{v}_4 & \hat{v}_9 \end{bmatrix} = 3 \begin{bmatrix} 21+14\overline{w}_n & 28+2w_n+9\overline{w}_n & 35+12c_n \\ 14+7\overline{w}_n & 21+6c_n & 28+2\overline{w}_n+9w_n \\ 7 & 14+7w_n & 21+14w_n \end{bmatrix}.$$

By elementary computations, one can determine the Bernstein–Bézier-form of all three biquadratic patches forming the first segment of the complex characteristic ring f. For $\lambda \in (1/4, 1)$,

$$\begin{aligned} f_0(1,1) = \frac{\hat{v}_3+\hat{v}_6+\hat{v}_7+\hat{v}_8}{4} &= p(\lambda) + c_n q(\lambda) \\ &:= (256\,\lambda^3 + 320\lambda^2 - 52\lambda - 2) + c_n\left(96\lambda^2 + 12\lambda\right). \end{aligned} \tag{6.19}$$

For $n \geq 3$, we have $c_n \geq -1/2$. Furthermore $p(\lambda) > 320\lambda^2 - 54\lambda - 2$ and $q(\lambda) > 0$ for all $\lambda \in (1/4, 1)$ so that

$$\begin{aligned} p(\lambda) + c_n q(\lambda) &> 320\lambda^2 - 54\lambda - 2 - (96\lambda^2 + 12\lambda)/2 \\ &= 272\lambda^2 - 60\lambda - 2 = 2(4\lambda - 1)(34\lambda + 1). \end{aligned} \tag{6.20}$$

That is, $f_0(1,1)$ is real and positive for $\lambda \in (1/4, 1)$. The eigenvector $\hat{v}$ and hence the characteristic ring f depends only on $\lambda = \hat{a}_1 = \hat{a}_{n-1}$ and on the valence n.

For $\lambda \in (1/4, 1)$, the minimum of the real parts of all Bernstein–Bézier coefficients is positive. Hence, by the convex hull property, the condition $c(u) \in \mathbb{R} \Rightarrow c(u) > 0$ in Theorem 5.24/105 is always satisfied. It remains to show regularity of the segment f_0 of the characteristic ring. The Jacobian determinant ${}^{\times}Df_0$ consists of three bicubic patches, which can also be expressed explicitly in Bernstein–Bézier-form. A careful analysis shows that all coefficients are positive if

$$p(\lambda) := 128\lambda^2(1-\lambda) - 7\lambda - 2 + 9\lambda c_n > 0. \tag{6.21}$$

By the convex hull property, this implies regularity of f_0. In particular, for $\lambda = 1/2$, we obtain $p(1/2) = 3/2\,(7 + 3c_n) > 0$ proving that the Doo–Sabin algorithm with standard weights is a C_1^1-algorithm. Figure 6.8/119 illustrates the situation: all Bernstein–Bézier coefficients of the Jacobian ${}^{\times}Df$ depend only on λ and c_n, and

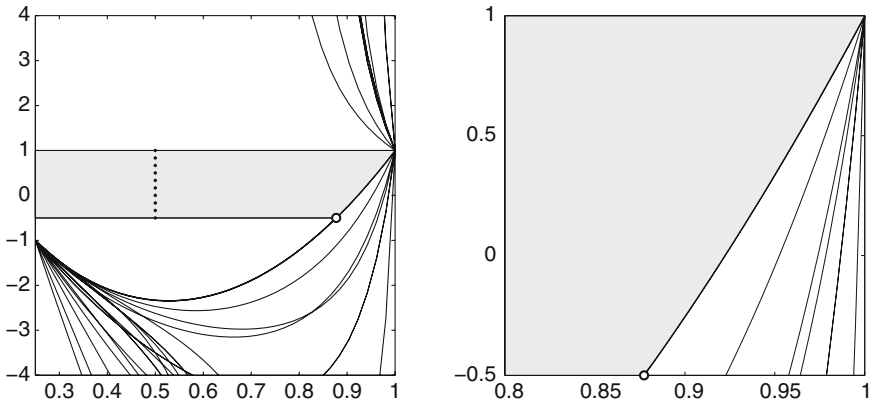

Fig. 6.8 Illustration of Theorem 6.2/119: (*left*) Admissible range of subdominant eigenvalues λ plotted in the (λ, c_n)-plane (see (6.21/118)) and (*right*) magnified detail.

they change sign on the lines plotted in the (λ, c_n)-plane. In particular, $p(\lambda) = 0$ for points on the thick line, and this line is bounding the shaded subset of the interval $(1/4, 1) \times [-1/2, 1)$. For given n, the eigenvalue λ yields a C_1^1-algorithm if and only if the point (λ, c_n) lies in this region. The dotted line, corresponding to the standard case $\lambda = 1/2$, indicates that this value is feasible for all values of n. Surprisingly, there is an upper bound $\lambda_{\sup}(n)$ with $p(\lambda) < 0$ for $1 > \lambda > \lambda_{\sup}(n)$. For such λ, ${}^{\times}Df$ actually reveals a change of sign, and the corresponding algorithm cannot be C_1^1. Fortunately, the upper bounds are quite close to 1 so that they do not impose severe restrictions when designing variants on the standard Doo–Sabin algorithm. More precisely, as indicated in Fig. 6.8/119 by the dot, the lowest upper bound occurs for $n = 3$. We have $c_3 = -1/2$ and

$$\lambda_{\sup}(n) \geq \lambda_{\sup}(3) = \frac{\sqrt{187}}{24} \cos\left(\frac{1}{3} \arctan\left(\frac{27\sqrt{5563}}{1576}\right)\right) + \frac{1}{3} \approx 0.8773.$$

The asymptotic behavior for $n \to \infty$ is

$$\lambda_{\sup}(n) \stackrel{*}{=} 1 - \frac{\pi^2}{7n^2}.$$

In summary, we have shown the following.

Theorem 6.2 (C_1^1-variants of Doo–Sabin subdivision). *Let $\hat{a}_0, \ldots, \hat{a}_{n-1}$ be the Fourier coefficients of a symmetric set of weights for the generalized Doo–Sabin algorithm. Then a standard algorithm is obtained if $\lambda := \hat{a}_1 = \hat{a}_{n-1}$ satisfies the condition*

$$1 > \lambda > \max\{1/4, |\hat{a}_2|, \ldots, |\hat{a}_{n-2}|\}.$$

The algorithm is C_1^1 if $p(\lambda) > 0$, and not C_1^1 if $p(\lambda) < 0$. In particular, the algorithm is C_1^1 when choosing the standard weights.

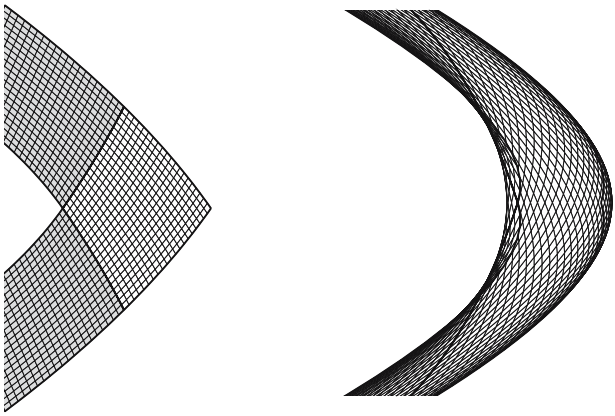

Fig. 6.9 Illustration of Theorem 6.2/119: Corner piece of the characteristic ring for $n = 3$ and subdominant eigenvalue (*left*) $\lambda = 0.5$ and (*right*) $\lambda = 0.95$. On the right hand side, the coordinate axes are scaled differently to clearly visualize non-injectivity.

6.3 Simplest Subdivision

When regarded as an algorithm for refining control meshes, one step of Simplest subdivision connects every edge-midpoint of the given mesh to the four midpoints of the edges that share both a vertex and a face with the current edge. For that reason, Simplest subdivision is sometimes also called *Mid-edge subdivision*. Once all midpoints are linked, the old mesh is discarded, as shown in Fig. 6.10/120. Thus

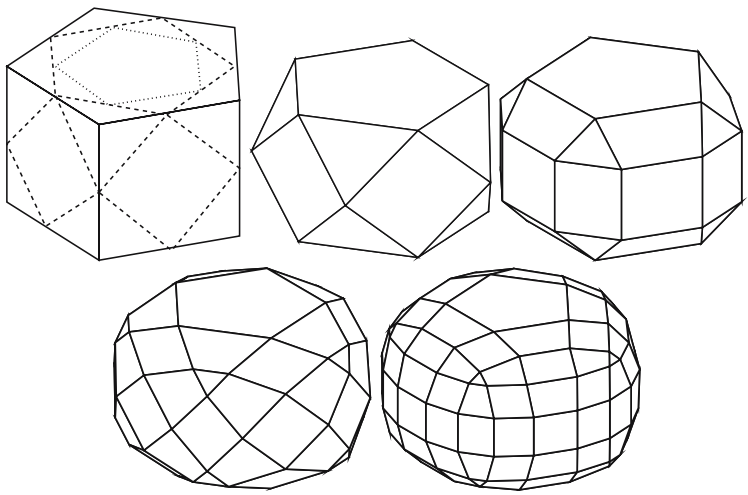

Fig. 6.10 Illustration of Simplest subdivision: Mesh refinement.

every new point, not on the global boundary, has exactly four neighbors; and every mesh point is replaced by a facet that is quadrilateral if the point is new. Each step can be interpreted as cutting off all vertices, along with neighborhoods that stretch half way to the neighbor vertex. The cuts are in general not planar.

The following feature justifies the name of the algorithm: all subdivision stencils are equal and of minimal size 2. The only weight used throughout is $1/2$ so that, in contrast to most other subdivision algorithms, there is no dependence on the valence n. Evidently, this setup is as simple as it can be.

To apply the analysis developed so far, we have to think of the algorithm not in terms of control meshes, but as a recursion for rings. To fit that pattern, we need to combine two steps of mesh refinement to generate a new ring $\mathbf{x}^{m+1}$ from the given ring $\mathbf{x}^m$. For that reason, Simplest subdivision is called a $\sqrt{2}$-algorithm. On a regular, quadrilateral control mesh with 4-valent vertices, a double step of mid-edge mesh refinement coincides with one subdivision step of the 4-direction box spline with directions $\Xi := \left[\begin{smallmatrix} 1 & 0 & -1 & 1 \\ 0 & 1 & -1 & -1 \end{smallmatrix}\right]$. Hence, the resulting limit surface is such a box spline. It is C^1, and each quadrilateral patch consists of four triangles of total degree 2, arranged in a quincunx pattern. From a combinatorial point of view, a double step of mid-edge mesh refinement for a general mesh coincides with one step of Doo–Sabin subdivision. That is, each n-gon is mapped to a smaller one. Using the same arrangement of control points $\mathbf{q}_\ell$ and weights $a = [a_0, \ldots, a_{n-1}]$ as in Fig. 6.7/117, we have

$$a_j = \begin{cases} \frac{1}{2} & \text{for } j = 0, \\ \frac{1}{4} & \text{for } j = 1, n-1, \\ 0 & \text{otherwise.} \end{cases}$$

The decisive point is that these weights are also used in the regular case $n = 4$. The structure of the Fourier blocks $\hat{A}_i$ of the subdivision matrix is the same as for the Doo–Sabin algorithm. But now, the weights in (6.17/117) are $p := 1/2, q := 1/4, r := 0$, and

$$\hat{a}_i = \sum_{j=0}^{n-1} w_n^{-ij} a_j = \frac{1 + \cos(2i\pi/n)}{2}, \quad i \in \mathbb{Z}_n.$$

For $i = 0, \ldots, n-1$, the non-zero eigenvalues of $\hat{A}_i$ are

$$\hat{a}_i \quad \text{and} \quad \frac{1}{4}, \frac{1}{4}.$$

The dominant eigenvalue of the subdivision matrix A is $\hat{a}_0 = 1$. Determining the subdominant eigenvalue is subtle here: If $n \geq 4$, we have $|\hat{a}_i| < |\hat{a}_1|$ for $i = 2, \ldots, n-2$, and also $1/4 < |\hat{a}_1|$ so that

$$\lambda := \hat{a}_1 = \hat{a}_{n-1} = \frac{1 + c_n}{2}, \quad c_n := \cos(2\pi/n),$$

is the double subdominant eigenvalue. That is, we obtain a standard algorithm. However, if $n = 3$, the upper left (4×4)-submatrices $\hat{A}'_i := \hat{A}_i(1:4, 1:4)$ of $\hat{A}_i$ read

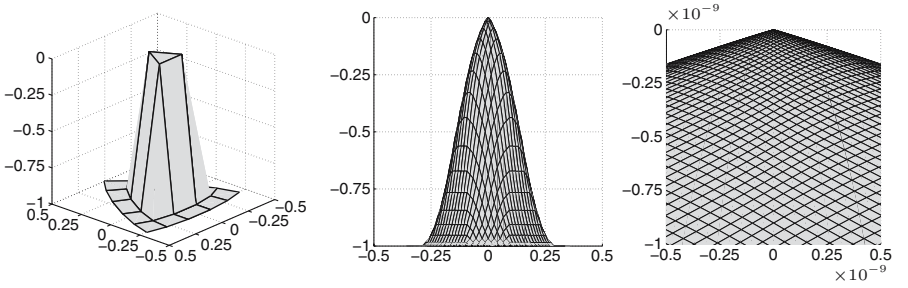

Fig. 6.11 Illustration of Simplest subdivision: (*left*) Input mesh, (*middle*) side view after three double-steps of refinement, and (*right*) close-up of visual cone tip after 20 double-steps of refinement. Nevertheless, from a mathematical point of view, the limit surface is smooth.

$$\hat{A}'_0 = \frac{1}{4}\begin{bmatrix} 4 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 \\ 3 & 0 & 0 & 1 \end{bmatrix}, \quad \hat{A}'_1 = \overline{\hat{A}'_2} = \frac{1}{8}\begin{bmatrix} 2 & 0 & 0 & 0 \\ 3-\mathbf{i}\sqrt{3} & 2 & 0 & 0 \\ 4 & 2 & 0 & 2 \\ 3+\mathbf{i}\sqrt{3} & 0 & 0 & 2 \end{bmatrix}.$$

The subdominant eigenvalue $\lambda = 1/4$ appears in multitude: its algebraic multiplicity in $\hat{A}'_0, \hat{A}'_1, \hat{A}'_2$ is $2, 3, 3$, respectively, while its geometric multiplicity in all three matrices is 2. Consequently, the Jordan decomposition of the subdivision matrix comprises the non-zero Jordan blocks

$$J_0 = 1, \quad J_1 = J_2 = \begin{bmatrix} 1/4 & 1 \\ 0 & 1/4 \end{bmatrix}, \quad J_3 = \cdots = J_6 = 1/4.$$

Thus, for $n = 3$, the algorithm is non-standard. Here, the more general theory, developed in Sect. 5.3/89, applies. The subdivision matrix has type $A \in \mathcal{A}_1^1$, and the characteristic ring ψ is defined according to Definition 5.10/92. The complex eigenvector defining ψ, arranged in matrix form as shown in Fig. 6.7/117, is

$$\hat{v} := \begin{bmatrix} 6+4\,c_n & 8+3\,c_n & 10+4\,c_n \\ 4+2\,c_n & 6+2\,c_n & 8+3\,c_n \\ 2 & 4+2\,c_n & 6+4\,c_n \end{bmatrix} + \mathbf{i}s_n \begin{bmatrix} 4 & 2 & 0 \\ 2 & 0 & -2 \\ 0 & -2 & -4 \end{bmatrix}.$$

As detailed in [PR97], regularity and uni-cyclicity can be verified using the same techniques as described above for the Doo–Sabin algorithm. Hence, we can state

Theorem 6.3 (Simplest subdivision is C_1^1). *Simplest subdivision is a C_1^1-algorithm for all valences $n \geq 3$.*

It remains to touch on the following two subjects: First, unlike tensor product B-splines, the nodal functions of box spline spaces do not always form bases. In particular, the system G of generating rings of Simplest subdivision is linearly dependent. But fortunately, the matrix A as specified here is a subdivision matrix in the sense that it does not have any ineffective eigenvectors.

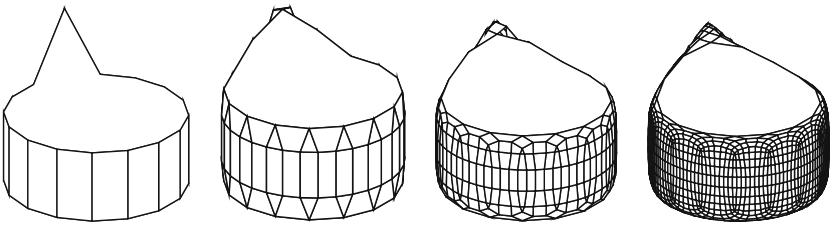

Fig. 6.12 Illustration of Simplest subdivision: Fast convergence for 3-sided facets and slow convergence for large facets.

Second, it must be noted that simplest subdivision, as described here, reveals serious shape artifacts for $n = 3$, and extremely slow convergence for high valences. In Fig. 6.11/122 *left*, we see an input mesh including a triangle, corresponding to an extraordinary point of valence $n = 3$. The first two coordinates correspond to the characteristic spline, while the third one has initial data which are 0 for the three innermost coefficients, and -1 otherwise. At the center, the surface seems to have a cone point, even if we zoom in by a factor of one billion to visualize the control mesh after 20 double steps of refinement (see Fig. 6.11/122, *right*). By the standards of Computer Graphics, this is hardly considered a smooth surface.[2] The apparent inconsistence between the theoretical result and the practical realization can be explained as follows: In the tangent plane, the behavior of rings is governed by the factor $\lambda^{m,1} = m\lambda^{m-1}$, while the component perpendicular to it decays as λ^m. To put it differently, let us consider the subdivision step from ring $\mathbf{x}^m$ to $\mathbf{x}^{m+1}$. Asymptotically, the tangential components are multiplied by $(1 + 1/m)/4$ and the normal component is multiplied by $1/4$, what does not make too much of a difference. As a consequence, the surface locally resembles the tip of a cone. Only for *very* large values of m, the slightly slower decay of the normal component prevails, and forces the rings to approach the tangent plane[3].

Another problem is depicted in Fig. 6.12/123. We see an input mesh which, after the first step, consists of triangles, quadrangles, and a 16-gon. The obvious problem concerns the extremely slow shrinkage of the 16-gon. It is due to the corresponding subdominant eigenvalue $\lambda \approx 0.962$, which is only slightly smaller than 1. As a consequence, a *very* large number of subdivision steps is required to obtain a mesh which is sufficiently dense for, say, visualization.

For practical purposes, Simplest subdivision should be modified, for instance by the Doo–Sabin weights (6.15/116) for $n \neq 4$. The modest increase in complexity is easily compensated for by superior shape properties.

---

[2] Imagine that the middle part of Fig. 6.11/122 is the size of Mount Everest. Then the detail on the right hand side is smaller than the breadth a hair. But mathematicians think in different categories.

[3] A quite instructive univariate analog of this case, justified by identifying $u = \lambda^m$, is given by the curve $c(u) = \big[u \ln |u|, |u|\big], u \in (-1/2, 1/2)$. Although the x-axis is easily verified to be the tangent at the origin, the image of the curve suggests a kink. The reader is encouraged to generate plots of the curve and its curvature at different scales.

Anyway, the latter observations clearly show that it is not sufficient to classify a subdivision algorithm as C_1^k to ensure fairness of the generated surfaces. Rather, an exacting analysis is necessary to scrutinize shape properties of subdivision surfaces. The next chapter focuses on that subject.

Bibliographic Notes

1. In the literature, the terms 'mask' and 'stencil' are not used in a consistent way. By our understanding, the entries in a row of the subdivision matrix, gathering contributions of the given control points to a single new one, can be displayed as a diagram called *stencil*. Conversely, a column of the subdivision matrix indicates how a single given control point influences the different new ones. This 'scattering' of contributions is called *mask*.

2. The spectrum of both the Catmull–Clark algorithm [CC78] and the Doo–Sabin algorithm [DS78] were first analyzed by Doo and Sabin [DS78] in order to assess and improve shape. In [BS86] an (almost correct) characterization of admissible weights α, β, γ for the Catmull–Clark algorithm is given.

3. A complete analysis of variants of the Doo–Sabin algorithm and of the Catmull–Clark algorithm can be found in [PR98]. Additional variants of the Catmull–Clark algorithm that modify rules not only for the central coefficient but also rules influenced by the central coefficient are proposed by Augsdörfer, Dodgson and Sabin [ADS06].

4. Simplest subdivision was published and analyzed by Peters and Reif [PR97] and also by Habib and Warren [HW99]. It is based on the Zwart–Powell element [Zwa73].

5. The smoothness properties of a number of other algorithms have been established. Loop derived the central stencil [Loo87] by analyzing the range of the eigenvalues. The analysis of Loop's algorithm is taken one step further in Schweitzer's thesis [Sch96] and complete analyses can be found in Umlauf's and Zorin's theses [Uml99, Zor97]. The interpolatory *Butterfly algorithm* was defined by Dyn, Gregory and Levin in [DGL90] and analyzed by Zorin in [Zor00a]. A bicubic interpolatory subdivision was introduced by Kobbelt [Kob96a] and analyzed in [Zor00a]. Kobbelt's surprising, since non-polynomial, $\sqrt{3}$-subdivision was derived and analyzed in [Kob00]. Velho and Zorin derived and analyzed the 4-8-algorithm [VZ01], an algorithm based on 4-direction subdivision. Leber's analysis [Leb94] was the first for a non-polynomial, bivariate interpolatory subdivision algorithm. While today, all relevant algorithms have been thoroughly analyzed, the algorithm by Qu and Gregory [QG92, GQ96b] still awaits a detailed treatment.

6. Zorin has developed and implemented automated machinery for proving C_1^k-regularity of subdivision algorithms [Zor00a].