

A bivariate C^1 subdivision scheme based on cubic half-box splines

Pieter Barendrecht^a, Malcolm Sabin^b, Jiří Kosinka^{a,*}

^a*Bernoulli Institute, University of Groningen, the Netherlands*

^b*Ely, United Kingdom*

Abstract

Among the bivariate subdivision schemes available, spline-based schemes, such as Catmull-Clark and Loop, are the most commonly used ones. These schemes have known continuity and can be evaluated at arbitrary parameter values. In this work, we develop a C^1 spline-based scheme based on cubic *half-box splines*. Although the individual surface patches are triangular, the associated control net is three-valent and thus consists in general of mostly hexagons. In addition to introducing stencils that can be applied in extraordinary regions of the mesh, we also consider boundaries. Moreover, we show that the scheme exhibits *ineffective eigenvectors*. Finally, we briefly consider architectural geometry and isogeometric analysis as selected applications.

Keywords: bivariate subdivision, three-valent meshes, honeycomb scheme, eigenanalysis

1. Introduction

The first publications on bivariate subdivision appeared 40 years ago ([Doo and Sabin, 1978](#); [Catmull and Clark, 1978](#)). These works introduced spline-based subdivision schemes generalising the subdivision of uniform biquadratic and bicubic tensor product B-spline surfaces to surfaces of arbitrary manifold topology, and techniques for understanding what happens at extraordinary vertices. Since then, a multitude of other schemes has been proposed, such as schemes based on three- and four-directional box splines ([Loop, 1987](#); [Peters and Reif, 1997](#); [Velho and Zorin, 2001](#); [Dodgson et al., 2009](#)), and schemes that are *not* spline based, including interpolatory schemes ([Dyn et al., 1990](#)) and the $\sqrt{3}$ scheme ([Kobbelt, 2000](#)). For a more complete overview, we refer the reader to one of the surveys ([Ma, 2005](#); [Cashman, 2012](#)) or books ([Warren and Weimer, 2001](#); [Peters and Reif, 2008](#); [Andersson and Stewart, 2010](#)) on subdivision surfaces.

In this work, we develop a bivariate C^1 subdivision scheme based on cubic *half-box splines*. It is a so-called *honeycomb scheme* as it subdivides three-valent meshes, which means that the ordinary regions of these meshes consist of hexagons. An overview of existing honeycomb schemes is discussed in Section 2.

The *limit surface* that results after a theoretical infinite number of subdivision steps is in our case composed of triangular half-box spline patches in the ordinary regions and infinite sequences of *spline rings* in the extraordinary regions. Details on half-box splines are reviewed in Section 3. The development of suitable *stencils* that can be used in extraordinary regions is considered in Section 4. The commonly used strategy of eigenanalysis also reveals that in the ordinary case, the scheme exhibits multiple *ineffective eigenvectors* ([Reif, 1998](#)). In addition, we discuss an approach for ordinary boundaries, and consider the evaluation of the limit surface using an extension of Stam's algorithm ([Stam, 1998b](#)).

We then take a more pragmatic look at our subdivision scheme, consider the generation of three-valent meshes, and discuss selected applications including architectural geometry and isogeometric analysis in Section 5. We conclude the paper in Section 6.

*Corresponding author

Email address: j.kosinka@rug.nl (Jiří Kosinka)

2. Related work

Although most work on bivariate subdivision focuses on triangular or quadrilateral meshes (or a combination thereof), a number of publications considers subdivision of three-valent meshes either as their main topic or as an illustrative example. As mentioned in [Claes et al. \(2002\)](#), hexagonal meshes can be subdivided using various factors, including 3, 4 and 7. Naturally, higher factors are also possible. However, the higher the factor, the faster the number of n -gons grows upon subdivision, which therefore makes it less attractive. Like most *honeycomb schemes*, [Claes et al. \(2002\)](#) construct a scheme that generates 3 new hexagons (or n -gons, in general) for each original one, and can be interpreted as a dual of the $\sqrt{3}$ scheme. The standard subdivision tools are used to tune the stencils and verify the necessary properties of the characteristic map. Boundary rules are added in [Beets et al. \(2002\)](#). Likewise, [Akleman and Srinivasan \(2002\)](#) propose a factor-3 honeycomb scheme using the same topological approach but with different stencils. The related paper ([Akleman et al., 2004](#)) considers the subdivision of three-valent pentagonal meshes.

The above schemes are not spline-based, but instead have subdivision stencil tuning as starting point. In contrast, a scheme based on half-box splines starts with the subdivision of ordinary regions of the surface and extends from there. Stencils for subdividing the ordinary regions of half-box spline surfaces are well-known ([Prautzsch and Boehm, 2002](#)). Extraordinary regions can be either approached using a hole-filling approach (such as the method described in [Prautzsch and Umlauf \(1999\)](#)), or using subdivision, as aimed for in this work.

Interestingly, [Oswald and Schröder \(2003\)](#) briefly describe an approach on how to handle extraordinary regions in a half-box spline scheme using a composition of subdivision operations in one of its examples, but does not go into detail. [Dodgson \(2005\)](#) also mentions the theoretical existence of a half-box spline scheme in a systematic overview of bivariate subdivision schemes. Finally, [Dyn et al. \(1992\)](#) and in particular [Dyn et al. \(2003\)](#) both consider factor-4 honeycomb schemes. These are however not spline based, although they use the same topological split.

Summarising, there is a small body of literature considering the use of three-valent meshes and the subdivision thereof. We are not aware of published work discussing the details of tuning stencils (such as the degrees of freedom) for extraordinary regions or boundaries in a half-box spline context, nor any discussions on the peculiarities of such a scheme, including the occurrence of ineffective eigenvectors. This paper focuses on both aspects, which form, together with the selected applications, our contributions.

3. Half-box splines

Half-box splines were introduced in [Frederickson \(1971\)](#), [Sabin \(1977\)](#) and [Prautzsch \(1984\)](#). Here, we consider the following approach. Starting with an indicator function (i.e., a piecewise constant function) $T(\mathbf{u})$ with an equilateral triangle as support, we place it on an isometric grid and consecutively convolve it in its three directions ξ_l , $l \in \{1, 2, 3\}$. Note that $\mathbf{u} = (u, v, w = 1 - u - v)$ are the barycentric coordinates over the triangle. This yields the functions $M_{\Xi}(\mathbf{u})$,

$$M_{\Xi}(\mathbf{u}) = \int_0^1 M_{\Xi \setminus \xi_l}(\mathbf{u} - \tau \xi_l) d\tau, \quad (1)$$

with Ξ the direction matrix containing direction vectors ξ_l . Counting the multiplicity of each ξ_l in Ξ as i, j, k allows for a shorthand notation $M_{ijk}(\mathbf{u})$ ([De Boor et al., 1993](#)). It follows that $M_{000}(\mathbf{u}) \equiv T(\mathbf{u})$.

After convolving once in all three directions, we obtain $H(\mathbf{u}) \equiv M_{111}(\mathbf{u})$. The resulting function is a symmetric (with respect to the directions) piecewise cubic function of which the pieces connect with C^1 continuity ([Prautzsch and Boehm, 2002](#)). Note that each additional application of convolution in all three directions raises the degree of the resulting half-box spline by three and its continuity by two. A schematic view of this process of *directional convolution* is shown in [Figure 1](#).

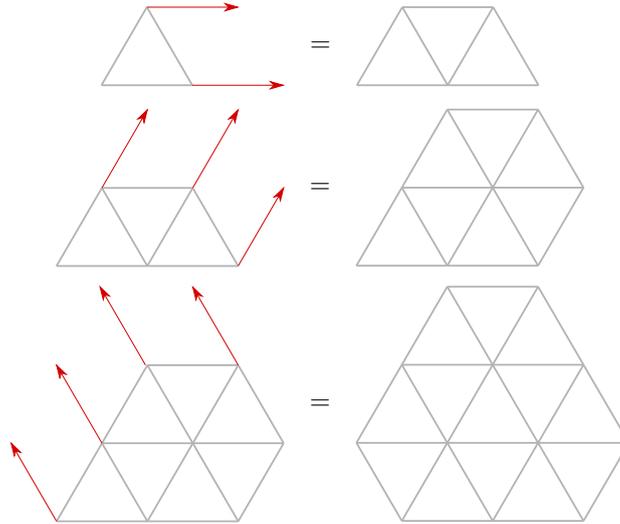


Figure 1: Directional convolution of an indicator function (top left) on an isometric grid in its three directions ξ_k (in red). The support of the resulting functions is shown (in grey).

3.1. Half-box spline patches and surfaces

Using the Bernstein-Bézier (BB) representation of the resulting functions $M_{\Xi}(\mathbf{u})$ (Farin, 1982; Boehm, 1983), it is straightforward to see that the pieces form a partition of unity. It then follows that we can construct spline patches that are affine invariant. This means that we can take each piece of $H(\mathbf{u})$ together with the centroid (i.e., the Greville abscissa) of $H(\mathbf{u})$ and overlap them. The result is a layout of the control net of a triangular cubic spline patch, as shown in Figure 2. These 13 control points — each one corresponding to a piece of $H(\mathbf{u})$ — can now be connected to form either two separate triangular nets, or a single hexagonal net. The latter is more convenient to work with, and is the preferable choice.

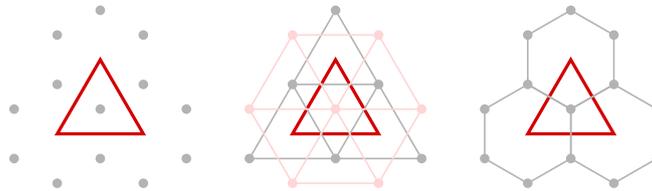


Figure 2: Constructing the control net of a triangular cubic half-box spline patch. The control points (left) can be connected to either form two separate triangular nets (middle), or a single hexagonal net (right).

Naturally, the BB-coefficients also allow a surface patch to be rendered as a triangular cubic Bézier patch. The control points for this patch also follow from the BB-coefficients, and are shown in Figure 3.

At this point, we can associate purely hexagonal meshes with C^1 piecewise cubic surfaces. However, in order to support surfaces of arbitrary topology, we have to allow vertices with a valency $n \neq 3$, or polygons that are not hexagons. Our approach focuses on subdivision, which we describe in the next section.

4. Subdivision and eigenanalysis

The starting point for spline-based subdivision is the *two-scale relation*, which expresses the spline function — in our case $H(\mathbf{u})$ — as a weighted sum of shifted and scaled (and possibly rotated) versions of itself. This is possible, because the support of the initial function — in our case $T(\mathbf{u})$ — is finitely *self-tessellating* (see e.g. Peters (2014)). In 2D, such shapes are referred to as *rep-tiles*. Like most schemes, we focus on *binary* subdivision, meaning that we consider the scaling factor $\frac{1}{2}$.

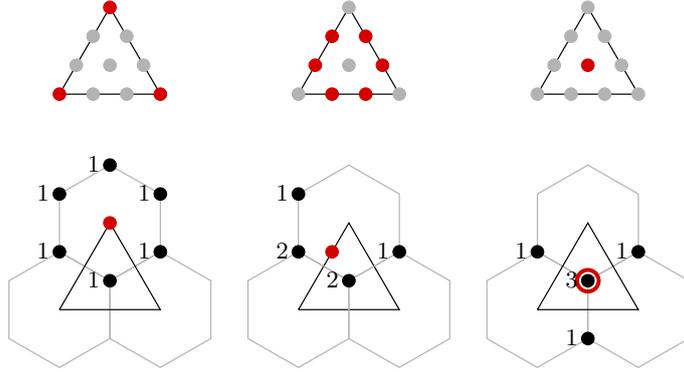


Figure 3: The Bézier stencils for obtaining the control points to represent a surface patch as a triangular cubic Bézier patch. As is customary, normalisation (in this case by $\frac{1}{6}$) of the coefficients is implied.

In order to obtain the coefficients for the two-scale relation, collectively referred to as the *subdivision mask*, we repeat the directional convolution process with $T(\mathbf{u})$ expressed as the sum of four indicator functions on equilateral triangles that are half the original size, one of which is rotated. Note that also the unit pulses in the directions ξ_i are now composed of two scaled unit pulses. We apply the convolution again in all three directions, and keep track of how many resulting scaled functions overlap at each triangle of the scaled grid. The resulting coefficients are multiplied by the scaling factor $\frac{1}{2}$ at each step of the convolution process, yielding a factor of $(\frac{1}{2})^3 = \frac{1}{8}$ in our case. This ultimately results in the subdivision mask. The process is illustrated in Figure 4.

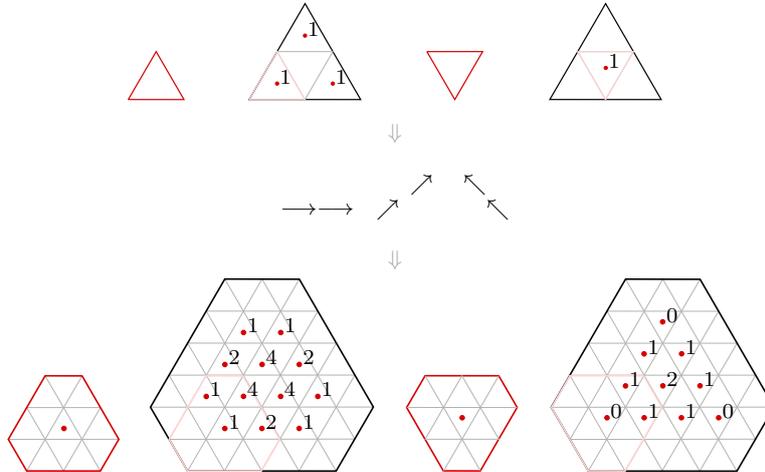


Figure 4: The two-scale relation for half-box splines. The initial function $T(\mathbf{u})$ (i.e., the indicator function with the equilateral triangle as support) is composed of shifted and scaled versions of itself, one of which is rotated (top). By convolving these with the composed unit pulse in all three directions ξ_i (see Figure 1), we obtain the mask for $H(\mathbf{u})$ (bottom). As is customary, scaling of the mask coefficients is implied.

Next, we consider a general half-box spline surface defined as

$$S(\mathbf{u}) = \sum_i P_i H_i(\mathbf{u}), \quad (2)$$

where \mathbf{P}_i are the control points of the ordinary hexagonal mesh. Each $H_i(\mathbf{u})$ can now be expressed as

$$H_i(\mathbf{u}) = \sum_j c_j h_{ij}(\mathbf{u}), \quad (3)$$

with c_j the mask coefficients from Figure 4 and $h_{ij}(\mathbf{u})$ the shifted and scaled copies of $H_i(\mathbf{u})$. An important observation is that the functions $h_{ij}(\mathbf{u})$ contribute to more than one $H_i(\mathbf{u})$, generally with different coefficients c_j , which is readily seen when illustrating the above on an isometric grid. We can then combine equations (2) and (3) so that each $h_{ij}(\mathbf{u})$ appears only once, multiplied by a sum of \mathbf{P}_i , each one in turn multiplied by a mask coefficient c_j . These terms can be interpreted as affine (and actually convex) combinations of the \mathbf{P}_i , as the involved c_j always sum to one (this holds because the h_{ij} also form a partition of unity). These affine combinations are the *subdivision stencils*, and can thus be used to subdivide purely three-valent hexagonal meshes, yielding a denser control net with every step which in the limit converges to the half-box spline surface. The cubic half-box spline yields (up to rotational symmetry) two stencils, which are shown in Figure 5.

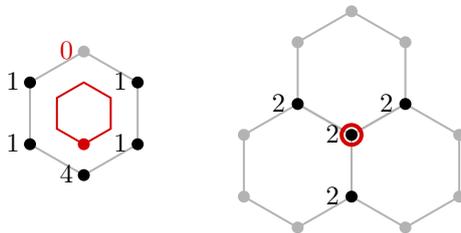


Figure 5: The two stencils \mathcal{S}_1 (left) and \mathcal{S}_2 (right) for subdividing (the control net of) a C^1 cubic half-box spline surface. These are readily extracted from the mask shown in Figure 4.

Rephrasing the above, the stencils follow from the mask, which can also be observed when writing the above expressions in matrix notation — the rows in the coefficient matrix are shifted repetitions of the mask, whereas the columns are the stencils. Finally, we note that there is a variety of other methods to deduce a subdivision mask, including the z -transform (Dyn, 2002), Fourier transform (Lai and Schumaker, 2007), and the arrow method (Dodgson et al., 2009), which is a graphical representation of discrete convolution.

4.1. Generalising to arbitrary topology

As mentioned in Section 3, in order to support surfaces of arbitrary topology, we should allow *extraordinary vertices* or *extraordinary faces* in the mesh. As this scheme is in a certain sense *dual* to Loop’s scheme (in the way that Doo-Sabin is dual to Catmull-Clark), we focus on the latter, and allow three-valent meshes with arbitrary n -gons. It should be noted that in practice this mostly means the addition of pentagons and heptagons, which are the equivalents of 5-valent and 7-valent extraordinary vertices in Loop’s scheme.

The derivation of appropriate subdivision stencils that can be applied to meshes of arbitrary topology is well understood (Peters and Reif, 2008). In our case, an extraordinary face exclusively surrounded by hexagons is considered. Figure 6 shows the example that is used in the forthcoming derivation.

In order to describe how the immediate environment of this n -gon can be subdivided, a *subdivision matrix* \mathcal{A}_n is composed of both the ordinary and extraordinary stencils. As the latter are unknown at this point, they appear in the subdivision matrix in symbolical form, as illustrated in Figure 7.

The main approach is then to perform *eigenanalysis* on the matrix. Because of the cyclic structure of the considered mesh, the subdivision matrix is *block-circulant* and as such allows analysis using the discrete Fourier transform, which turns \mathcal{A}_n into a *block diagonal* matrix \mathcal{D}_n . In matrix notation, we have

$$\mathcal{A}_n = \begin{pmatrix} A_1 & A_2 & \dots & A_n \\ A_n & A_1 & \dots & A_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_2 & A_3 & \dots & A_1 \end{pmatrix}, \quad (4)$$

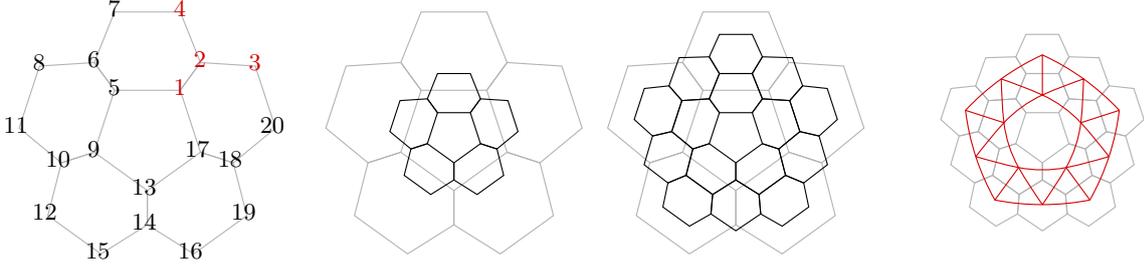


Figure 6: An n -gon in an otherwise hexagonal mesh (left). The square subdivision matrix \mathcal{A}_n (containing coefficients that are currently unknown) can be used to subdivide the mesh (mid-left). An extended subdivision matrix $\bar{\mathcal{A}}_n$ is used to subdivide a larger neighbourhood (mid-right) that is required for the evaluation of surface patches, which form the well-known spline rings around the limit position of the n -gon (right).

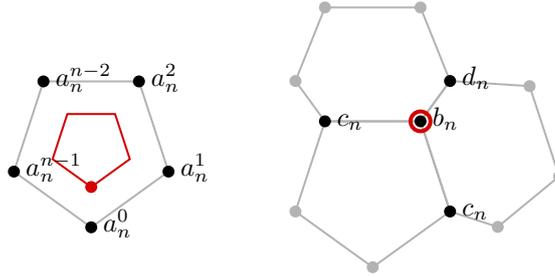


Figure 7: Generalisations of the stencils \mathcal{S}_1 and \mathcal{S}_2 as shown in Figure 5 with symbolical coefficients. The example considers $n = 5$ but the same approach is used for all $n \geq 3$.

where A_i are blocks of size 4×4 . In order to obtain \mathcal{D}_n , we first take the Kronecker product of the discrete Fourier transform (DFT) matrix F_n , where n is the valency of the n -gon, with the unit matrix of the appropriate size (here, 4), which yields a block-DFT matrix $\mathcal{F}_{n,4} = F_n \otimes I_4$. We then obtain

$$\mathcal{D}_n = \mathcal{F}_{n,4}^{-1} \mathcal{A}_n \mathcal{F}_{n,4} = \begin{pmatrix} D_0 & & & \\ & D_1 & & \\ & & \ddots & \\ & & & D_{k-1} \end{pmatrix}. \quad (5)$$

For the *right* eigenvectors w of \mathcal{D}_n we have

$$\begin{aligned} \mathcal{D}_n w &= \lambda w \\ \mathcal{F}_{n,4}^{-1} \mathcal{A}_n \mathcal{F}_{n,4} w &= \lambda w \\ \mathcal{A}_n \mathcal{F}_{n,4} w &= \lambda \mathcal{F}_{n,4} w \\ \mathcal{A}_n v &= \lambda v, \end{aligned}$$

where $v = \mathcal{F}_{n,4} w$. It follows that the matrix \mathcal{D}_n is *similar* to \mathcal{A}_n . That is, they share the same *eigenvalues*, but in general have different *eigenvectors*. The right eigenvectors of \mathcal{D}_n are the right eigenvectors of the blocks D_j vertically padded with zeroes.

Using the ordering of control points as shown in Figure 6 and the notation introduced in Figure 7, the blocks A_i now follow as

$$A_1 = \begin{pmatrix} a_n^0 & 0 & 0 & 0 \\ b_n & d_n & 0 & 0 \\ \frac{1}{2} & \frac{1}{8} & \frac{1}{8} & 0 \\ \frac{1}{2} & \frac{1}{8} & 0 & \frac{1}{8} \end{pmatrix}, A_2 = \begin{pmatrix} a_n^1 & 0 & 0 & 0 \\ c_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & 0 & 0 \end{pmatrix}, A_3 = \begin{pmatrix} a_n^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \dots, A_{n-1} = \begin{pmatrix} a_n^{n-2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, A_n = \begin{pmatrix} a_n^{n-1} & 0 & 0 & 0 \\ c_n & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Note that as $i \in [1, n]$, this covers faces of any valency n . The red zeroes in A_2 and A_n come from the ‘skipped’ vertex in stencil \mathcal{S}_1 (see Figure 5) but have no further significance. The blocks D_j , with $j \in [0, n - 1]$, can then be expressed as

$$D_j = \sum_{i=0}^{n-1} \omega^{ij} A_{i+1} = \begin{pmatrix} \sum \omega^{ij} a_n^i & 0 & 0 & 0 \\ b_n + 2c_n \cos\left(\frac{2\pi j}{n}\right) & d_n & 0 & 0 \\ \frac{1}{2} + \omega^{-j} \frac{1}{8} & \frac{1}{8} + \omega^{-j} \frac{1}{8} & \frac{1}{8} & 0 \\ \frac{1}{2} + \omega^j \frac{1}{8} & \frac{1}{8} + \omega^j \frac{1}{8} & 0 & \frac{1}{8} \end{pmatrix}.$$

Observe that as all blocks A_i are *lower triangular*, the blocks D_j are as well. This means that the eigenvalues of \mathcal{D}_n (and therefore of \mathcal{A}_n) follow directly as the union of diagonal entries of the blocks D_j . This considerably simplifies the tuning of stencil coefficients a_n^i and d_n , as these directly correspond to eigenvalues. We can now use the well-known result from subdivision surface theory (Peters and Reif, 2008) that for a scheme to be C^1 , the first set of necessary conditions are a single dominant eigenvalue of 1 (which is satisfied by the top-left eigenvalue in D_0) and two *subdominant* eigenvalues from the blocks D_1 and D_{n-1} . These subdominant eigenvalues also indicate the *rate of contraction* of a subdivision scheme, which ideally should be $\frac{1}{2}$ for a binary scheme (Donatelli et al., 2019). From the top-left entries of the D_j it follows that

$$\lambda_j = \sum_{k=0}^{n-1} \omega^{jk} a_n^k, \text{ or } \boldsymbol{\lambda} = F_n \mathbf{a}, \quad (6)$$

and therefore $\mathbf{a} = F_n^{-1} \boldsymbol{\lambda}$. Choosing a spectrum of $\boldsymbol{\lambda} = [1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{4}, \frac{1}{2}]$ then gives

$$a_n^k = \frac{1}{n} \sum_{j=0}^{n-1} \omega^{-jk} \lambda_j = \frac{1}{4n} \sum_{j=0}^{n-1} \omega^{-jk} + \frac{3}{4n} + \frac{1}{4n} (\omega^{-k} + \omega^k) = \frac{1}{4n} \sum_{j=0}^{n-1} \omega^{-jk} + \frac{3}{4n} + \frac{2}{4n} \cos\left(\frac{2\pi k}{n}\right),$$

where all λ were replaced by $\frac{1}{4}$ in the first step and corrected for $j = \{0, 1, n-1\}$ with missing parts $[\frac{3}{4}, \frac{1}{4}, \frac{1}{4}]$. We can further simplify this, as the ω^{-jk} sum to n for $k = 0$ and to 0 otherwise. As such, we obtain

$$a_n^k = \begin{cases} \frac{1}{4} + \frac{5}{4n} & k = 0, \\ \frac{3+2 \cos\left(\frac{2\pi k}{n}\right)}{4n} & \text{otherwise,} \end{cases} \quad (7)$$

which is in fact the same expression as the one which appeared in Doo and Sabin (1978).

A minor drawback of this approach in the current context is that (7) does not match the stencil for the ordinary case as shown in Figure 5. Therefore, (7) is used for all $n \neq 6$, whereas the stencil shown in Figure 5 covers the ordinary case.

In the case of the second stencil, we have the choice of either using the original \mathcal{S}_2 for all n -gons, or to tune the coefficients b_n , c_n and d_n . Although using the former approach yields generally satisfactory results, tuning of the coefficients is considered in Appendix A.

The second set of necessary conditions for a scheme to be C^1 is that its *characteristic map*, that is, the limit surface of the mesh defined by the two right eigenvectors associated with the subdominant eigenvalues (also referred to as the *natural configuration*), should be regular and injective. Because of rotational symmetry and the scaling property of the spline rings composing the characteristic map, analysis of one sector of a single ring suffices (Warren and Weimer, 2001). Figure 8 shows characteristic maps for selected valencies when using the original \mathcal{S}_2 .

Although the characteristic map satisfies the required conditions for all depicted valencies, the natural configurations for $n = 3$ and $n = 4$ exhibit unusual behaviour. In both cases there are coinciding vertices, resulting in degenerate polygons. Furthermore, for $n = 3$ the natural configuration contains self-intersecting polygons. For applications where the subdivided mesh (as opposed to the limit surface) is the object of interest, this could lead to undesirable results. Appendix A considers a tuned version of the second stencil, which remedies these issues.

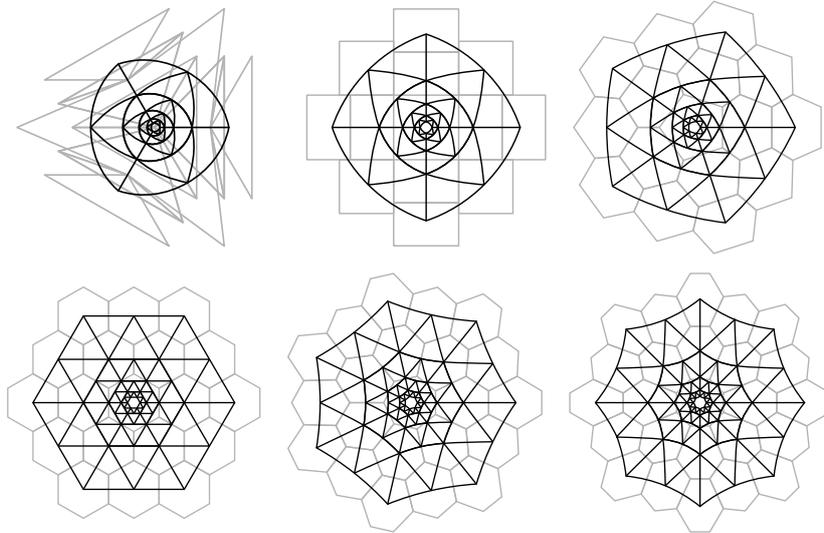


Figure 8: Natural configurations (grey) and characteristic maps (black) for $n = \{3, 4, 5, 6, 7, 8\}$.

4.2. Ineffective eigenvectors

A more recent addition to the toolbox of subdivision analysis is the notion of *ineffective eigenvectors* (Reif, 1998; Peters and Reif, 2008). Consider a general half-box spline *subdivision* surface, which can be expressed in a similar way to (2), but now contains a subdivision component:

$$\mathcal{S}(\mathbf{u}) = H^T(\mathbf{u})\mathcal{A}^k\mathbf{Q}, \quad (8)$$

with \mathbf{Q} a column of control points constituting the mesh that is subdivided k times using some appropriate square subdivision matrix \mathcal{A} . If we now diagonalize this subdivision matrix (or apply Jordan decomposition in case diagonalization is not possible), we obtain $\mathcal{A} = V\Lambda V^{-1}$, where V is a matrix with the right (generalised) eigenvectors v_i as columns. Rewriting (8) then yields

$$\mathcal{S}(\mathbf{u}) = H^T(\mathbf{u})V\Lambda^kV^{-1}\mathbf{Q}. \quad (9)$$

An eigenvector v_i is called *ineffective* if $H^T v_i = 0$ while its associated eigenvalue $\lambda_i \neq 0$. The consequence of an ineffective eigenvector is that a part of the spectrum of \mathcal{A} does not contribute to the resulting surface. This can pose a problem for the analysis of a subdivision scheme (as used in the preceding section) if eigenvectors associated with the dominant, subdominant or subsubdominant eigenvalues coming from the appropriate block(s) D_j are ineffective. Ineffective eigenvectors associated with other eigenvalues do not affect the analysis and can be considered harmless.

A possible approach is to check for each individual eigenvector v_j whether it is effective or not, and if it is not, remove it by modifying the subdivision matrix (Peters and Reif, 2008). However, it also suffices to verify whether the set of eigenvectors relevant for C^1 continuity mentioned above is effective.

For the half-box spline subdivision scheme, it turns out that there are three ineffective eigenvectors. To our knowledge, it is the first non-artificial scheme that exhibits this phenomenon. To understand the occurrence of these ineffective eigenvectors, note that the cubic triangular surface patches are defined using the 13 cubic pieces of $H(\mathbf{u})$, which means that the definition is *overcomplete*. After all, only 10 basis functions are required to span the complete cubic space (see Figure 3). The three ineffective eigenvectors are associated with eigenvalues $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{4}$, which looks problematic at first sight. However, eigenanalysis in the Fourier domain shows that these eigenvalues do not come from one of the relevant blocks D_j . The ineffective eigenvectors are therefore harmless in this case, although it results in the peculiar property that the half-box spline scheme (in the ordinary case $n = 6$) has a *triple* subdominant eigenvalue.

To conclude, we mention that something similar occurs in the *simplest* subdivision scheme (Peters and Reif, 1997). It is based on the four-directional piecewise quadratic Zwart-Powell box spline, which uses 7 pieces to define a quadratic triangular surface patch. This is also an overcomplete definition, as only 6 basis functions are needed to span the complete quadratic space. However, the eigenvector v_j that is in the kernel of the blending functions is associated with a zero-valued eigenvalue λ_j and is therefore not referred to as ineffective, as its eigenvalue already annihilates any contribution to the surface.

4.3. Boundaries and creases

Support for boundaries in a subdivision scheme, and likewise for (sharp) creases, significantly increases the versatility of the objects that can be modelled with it. Adding boundary rules to honeycomb schemes has been considered in Beets et al. (2002), and for selected three-directional box splines in Sabin and Bejancu (2003). In our case, we have cubic surface patches with C^1 connectivity, which hints at the use of cubic subdivision with double knots (Kosinka et al., 2013).

Figure 9 illustrates our approach at an ordinary boundary (i.e., a contiguous set of hexagons at the boundary of a three-valent mesh). The Bézier points (see Figure 3) controlling the boundary curve are highlighted in red, green and black. Using e.g. blossoming (Ramshaw, 1989), it is readily observed that the green points form the control polygon of a cubic B-spline curve with double knots. The subdivision scheme for such a curve, which can act as either a boundary or as a sharp crease, is known (Kosinka et al., 2013). Corners of the mesh (such as the black point) can be reflected in the B-spline curve by using a triple knot.

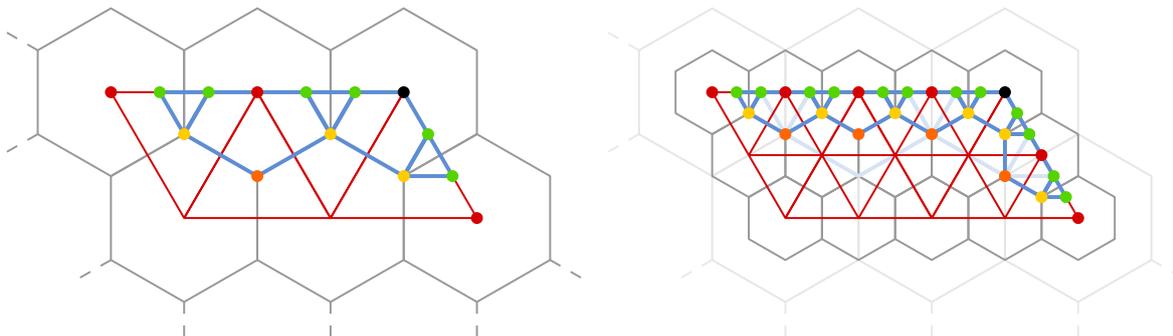


Figure 9: Original (left) and subdivided (right) hexagonal mesh (grey) with triangular surface patches (red). The red, green and black points are the Bézier points defining the boundary curves (which is piecewise C^1 cubic). At the boundary, the hexagons are truncated into pentagons and triangles to form an auxiliary structure (blue).

We use the control polygon of the B-spline to truncate the hexagons at the boundary. Rather than using *half-polygons* as used in Beets et al. (2002), we build an auxiliary structure composed of pentagons and triangles (blue), which is defined by the green and black Bézier boundary points and the interior points (yellow and orange). Showing this structure to the user instead of the contiguous set of hexagons at the boundary clearly provides a more intuitive preview of the eventual boundary curve associated with the mesh.

Direct interaction with the boundary control polygon proves difficult. Indeed, moving a single green point requires the update of other points of the boundary hexagon it lies within, which in turn influence the position of neighbouring green points. As such, interaction in this fashion quickly turns into solving a global system, which is not an acceptable approach. We defer handling interaction, along with handling the various types of corners (there are two convex and two concave configurations) to future research.

4.4. Evaluation of the limit surface

Using the terminology introduced above, extension of Stam’s algorithm (Stam, 1998b) to evaluate the limit surface at arbitrary parameter values \mathbf{u} is now straightforward. Taking the mesh with control points \mathbf{P}_i from Figure 6 (left) as a starting point, it can be *virtually* subdivided (i.e., locally subdivided in memory) an arbitrary number of $(l - 1)$ times using the matrix \mathcal{A}_n to yield $4n$ new control points, see Figure 6 (mid-left). This is followed by a single subdivision using the extended matrix $\tilde{\mathcal{A}}_n$, resulting in $9n$ control points

(Figure 6, mid-right), to allow evaluation of a surface patch in the desired spline ring (Figure 6, right). The relevant points for evaluation are extracted from the resulting set of points $\mathbf{Q} = \mathcal{A}_n \mathcal{A}_n^{l-1} \mathbf{P}$ using an *extraction* or *picking* matrix X_n^k , which are then converted to Bézier control points (see Figure 3) so that the surface can be evaluated at \mathbf{u} . Obtaining the correct level l and sector k for a given \mathbf{u} follows the same approach as in Stam (1998a), which uses the binary logarithm of u to determine l and subsequently uses v and w to determine k . Figure 10 shows the limit surface in the neighbourhood of a pentagon. Another limit surface is shown in Figure 11.

One of the key points of Stam’s algorithm is to diagonalise \mathcal{A}_n as $\mathcal{A}_n = V_n \Lambda_n V_n^{-1}$ in order to speed up the computation of $\mathcal{A}_n^{l-1} = V_n \Lambda_n^{l-1} V_n^{-1}$. However, as mentioned in Peters and Reif (2008), this only becomes viable from a relatively high subdivision level l .

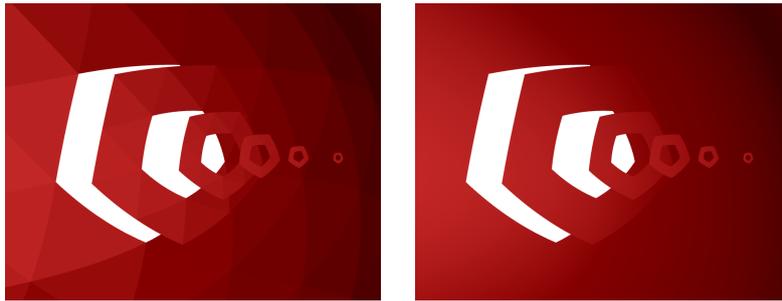


Figure 10: Limit surface with spline rings (offset) around an isolated pentagon in a three-valent mesh, flat-shaded (left) to show the individual half-box spline patches, and smooth-shaded (right).

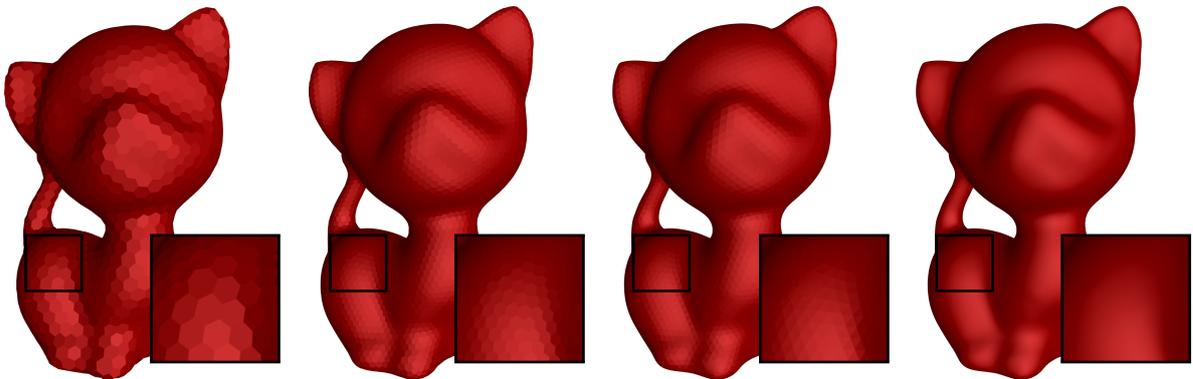


Figure 11: A three-valent mesh (left) followed by a step of our subdivision scheme (middle-left) and its corresponding limit surface, flat shaded per surface patch (middle-right) and smooth-shaded (right).

The evaluation of partial derivatives at arbitrary parameter values also follows the usual approach, which allows the computation of tangent and normal vectors of the surface.

Finally, we mention that the evaluation of limit surfaces near boundaries is a straightforward extension, which follows the methods described in Smith et al. (2004); Lacewell and Burley (2007), both based on ghost points and the Jordan normal form of subdivision matrices.

5. Our scheme in practice

In this section we look at our scheme from a more pragmatic point of view. We consider the generation of three-valent meshes and comment on the relevance of our scheme in two selected fields.

5.1. On three-valent meshes

Compared to the abundance of triangle and quad meshes, honeycomb meshes are relatively rare. Triangles are often the preferred shape to re-mesh dense (scanned) objects, whereas quads are commonly used in the digital design of objects.

There are a couple of operations that either allow the conversion of (triangle) meshes to three-valent ones or to generate such meshes from scratch. Starting from a triangle mesh, the *geometric dual* generates a three-valent mesh by connecting the vertex centroids of triangles over their edges. Another option is to trisect each edge of a triangle mesh and again connect the vertices to form a honeycomb mesh. Note that the result is topologically equivalent to the geometric dual of a $\sqrt{3}$ operation (Kobbelt, 2000). Finally, centroidal Voronoi tessellations (CVTs) (Du et al., 1999) can be used to construct three-valent meshes. These are Voronoi tessellations for which the generators coincide with the vertex centroids of the faces. In our case, the occurrence of short edges in the CVTs should be minimised, for which the approach described in Wang et al. (2018) can be used. Figure 12 illustrates the different approaches.

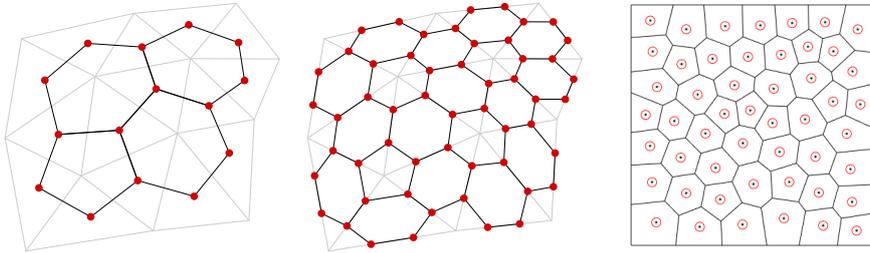


Figure 12: Generating the geometric dual (left) and trisection (centre) of a triangle mesh to obtain a three-valent mesh. Generating a centroidal Voronoi tessellation (right) also yields appropriate three-valent meshes.

5.2. Relevance in architectural geometry

Hexagons and three-valent meshes in general play an increasingly important role in architecture (Jiang et al., 2014), or more specifically, *architectural geometry* (Pottmann et al., 2015). Constructing or covering facades with panels forming a honeycomb mesh typically results in an appreciated organic look. Figure 13 shows some examples of freeform designs in architecture.

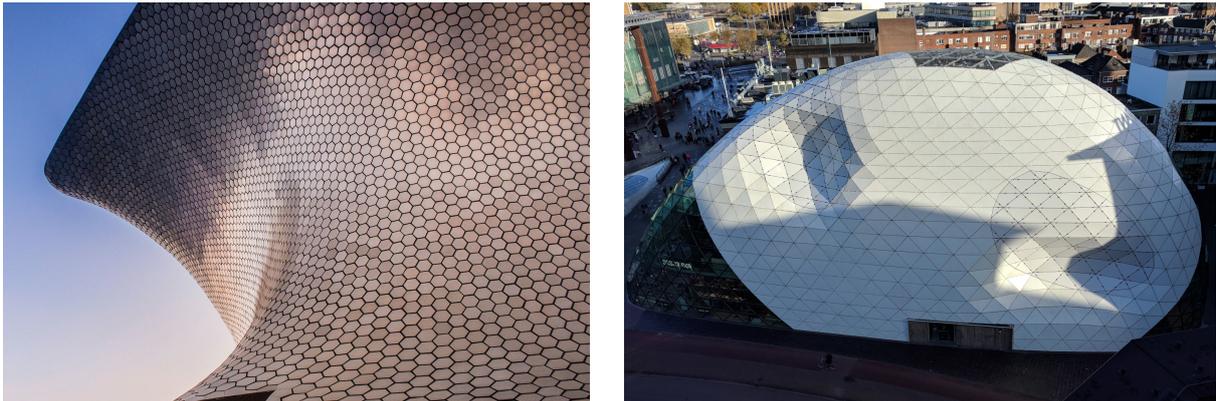


Figure 13: Museo Soumaya in Mexico City with a hexagonal facade (photo courtesy of Erick Aguilar Suárez) and Blob in Eindhoven, the Netherlands with a triangular facade.

The advantage of three-valent meshes in this context lies in the construction of the *nodes* connecting the panels, which are considerably simpler compared to their counterparts in meshes composed of triangular panels, where on average 6 panels are connected to a single node. The drawback of honeycomb meshes is

that the faces are generally not planar, something which currently poses a bottleneck in the manufacturing process of these panels.

Generation of hexagonal meshes with planar faces is discussed in Wang et al. (2008), which uses the concept of *tangent* dual meshes. A variation of our scheme could perhaps be developed, tuning the stencils in order to improve the planarity of the resulting faces.

5.3. Relevance in numerical methods

Our scheme could also be useful in the context of numerical methods such as isogeometric analysis (IgA), see e.g. Cirak et al. (2000); Wei et al. (2015). One of its advantages is the C^1 parameterization that follows from extending Stam’s algorithm. In contrast, Stam’s parameterization for Catmull-Clark and Loop is merely C^0 as the subdominant eigenvalues λ_j of these schemes are larger than $\frac{1}{2}$ for valencies higher than the respective regular values of 4 and 6. In the computation of partial derivatives, the term $(2\lambda_j)^{l-1}$ then becomes unbounded at the extraordinary vertex, and therefore the parameterization cannot be regarded as C^1 . This has implications for the (stability of) numerical integration of terms containing partial derivatives and/or Jacobians near the extraordinary vertex.

Another advantage of having $\lambda_j = \frac{1}{2}$ in this context is the uniform contraction of elements, which is important when the mesh is (adaptively) refined during a simulation.

Naturally, IgA could be applied directly to three-valent meshes, but as mentioned, these are not abundant. Another option is to start from an existing triangle mesh with the constraint that the limit surface should interpolate the vertices of that triangle mesh, and compute its *inverse* geometric dual, that is, the three-valent mesh whose geometric dual is the existing triangle mesh (see Figure 14).



Figure 14: The initial triangle mesh (centre) with its inverse geometric dual (left) and actual geometric dual (right).

As the subdivision limit positions of the n -gons in our scheme are in fact the centroids of those n -gons for all valencies $n \geq 3$ (which follows directly from the *left* eigenvector — also referred to as *limit stencil* — associated with the dominant eigenvalue), this appears to be straightforward. With the exception of anomalies, this computation leads to an underdetermined system of full column rank and an augmented matrix of the same rank. This proves the existence of inverse dual meshes in this context. Because there are an infinite number of solutions to the system, we could either minimise an objective function based on some metric (e.g. the planarity of the faces or angle between edges), or compute a minimum-norm solution with regard to a reference solution. In this work we have only explored the latter approach, taking the actual geometric dual of the existing triangle mesh as a reference solution. Unfortunately, the results are not always convincing for coarse meshes, which are typically the meshes used in numerical analysis. For somewhat denser meshes, such as the example shown in Figure 14, the approach works nicely.

Key aspects of finite element methods are numerical integration (quadrature) and adaptivity (local refinement), the latter strongly related to (local) linear independence. Although a detailed discussion of

these topics is out of scope, we remark that both topics have been studied in the context of subdivision splines and are areas of active research (see e.g. [Barendrecht et al. \(2018\)](#); [Peters and Wu \(2006\)](#); [Zore et al. \(2016\)](#)).

6. Discussion and conclusion

In this work we have completed a subdivision scheme based on cubic half-box splines by developing stencils for extraordinary polygons. The scheme meets the necessary and sufficient conditions to be overall C^1 , and has been checked for ineffective eigenvectors (which are present but are harmless in our case). As the scheme is spline-based, its limit surface and partial derivatives can be evaluated everywhere using a straightforward extension of Stam’s algorithm.

We discussed the (potential) relevance of honeycomb meshes and their subdivision for selected fields, which is promising but requires more extensive research.

Additional directions for future research include the topic of ineffective eigenvectors, a complete treatment of boundaries, and the derivation of other fraction-box spline subdivision schemes. We expect a four-directional half- or quarter-box spline scheme to exist with a control net composed of quads and octagons, making it dual to the 4-8 scheme ([Velho and Zorin, 2001](#)).

Appendix A. Tuning the second stencil

In this section we consider tuning \mathcal{S}_2 (in Loop’s scheme this would be equivalent to tuning the *edge stencil*). We note that a tuned version results in a subdivision scheme that is no longer *uniform*. This somewhat complicates the evaluation of the limit surface in extraordinary regions, as a wider neighbourhood around n -gons with $n \neq 6$ must be (virtually) subdivided before it can be evaluated using the standard approach.

As mentioned, the natural configurations for $n \in \{3, 4\}$ show undesired behaviour. Somewhat surprisingly, the associated characteristic maps still satisfy the required conditions, which we illustrate again in [Figure A.15](#). To do so, the vertices of the central n -gon in the natural configurations are set to $z = 1$ while keeping the other vertices at $z = 0$. Subsequently, the resulting mesh is rotated in 3D to allow for a better view of the characteristic rings when they are offset.

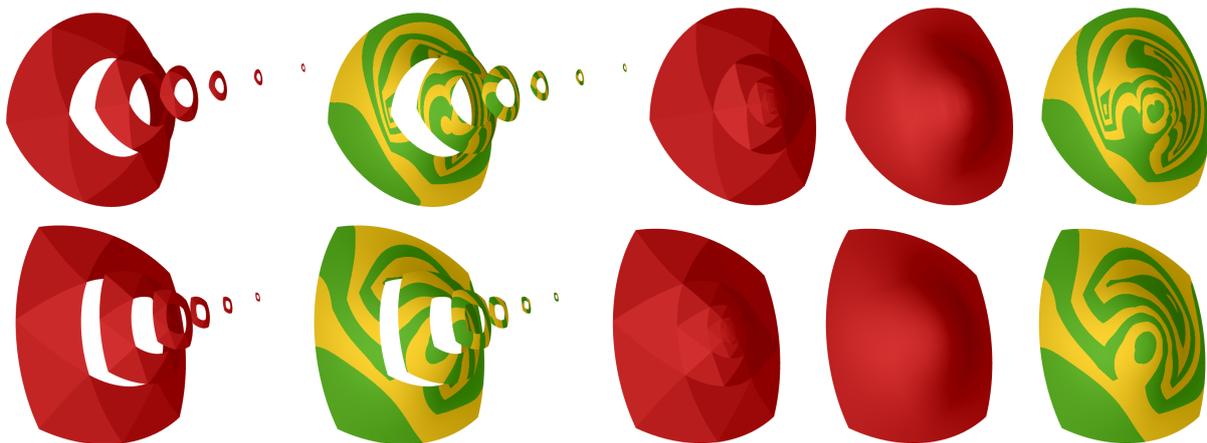


Figure A.15: Characteristic rings for $n = 3$ (top) and $n = 4$ (bottom) rendered using flat and smooth shading and with isophotes using the original \mathcal{S}_2 .

When tuning \mathcal{S}_2 , the condition for bounded curvature at the limit position of the n -gons comes into play, which states that the three *subsubdominant* eigenvalues μ should have the squared value of the subdominant eigenvalue *and* come from the blocks D_0 , D_2 and D_{n-2} . In our case, it directly follows that $\mu = \frac{1}{4}$. The

only way to satisfy this is to set $d_n = \mu$. Clearly, we also have the condition $b_n + 2c_n + d_n = 1$, which now yields the condition $b_n + 2c_n = \frac{3}{4}$. This leaves only one degree of freedom, which then theoretically results in a range of admissible values of b_n and c_n . Figure A.16 shows improved natural configurations and their characteristic maps using empirical values of $b_3 = 0.45$ and $b_4 = 0.35$.

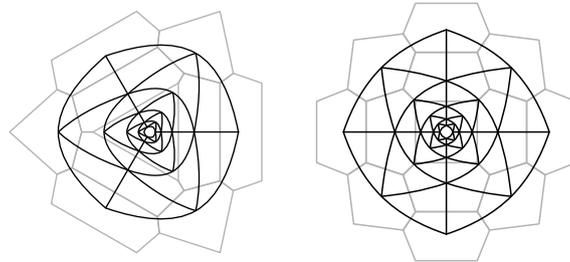


Figure A.16: Improved natural configurations and their characteristic maps for $n = 3$ (left) and $n = 4$ (right), cf. Figure 8.

To conclude, we mention the more pragmatic approach of applying the scheme with modified \mathcal{S}_2 only in the first subdivision step (which has the most influence over the eventual shape of the limit surface), after which the original \mathcal{S}_2 can be used again.

References

- Akleman, E., Srinivasan, V., 2002. Honeycomb subdivision, in: Proc. 17th Int. Symp. Computer & Information Sciences, pp. 137–141.
- Akleman, E., Srinivasan, V., Melek, Z., Edmundson, P., 2004. Semiregular pentagonal subdivisions, in: Shape Modeling Applications, 2004. Proceedings, IEEE. pp. 110–118.
- Andersson, L.E., Stewart, N.F., 2010. Introduction to the mathematics of subdivision surfaces. volume 120. Siam.
- Barendrecht, P.J., Bartoň, M., Kosinka, J., 2018. Efficient quadrature rules for subdivision surfaces in isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 340, 1–23.
- Beets, K., Claes, J., Van Reeth, F., 2002. Borders, semi-sharp edges and adaptivity for hexagonal subdivision surface schemes, in: Advances in Modelling, Animation and Rendering. Springer, pp. 151–166.
- Boehm, W., 1983. Generating the Bézier points of triangular splines, in: Surfaces in Computer Aided Geometric Design, North-Holland Amsterdam. pp. 77–91.
- Cashman, T.J., 2012. Beyond Catmull-Clark? A survey of advances in subdivision surface methods. Computer Graphics Forum 31, 42–61.
- Catmull, E., Clark, J., 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer-Aided Design 10, 350–355.
- Cirak, F., Ortiz, M., Schröder, P., 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. International Journal for Numerical Methods in Engineering 47, 2039–2072.
- Claes, J., Beets, K., Van Reeth, F., 2002. A corner-cutting scheme for hexagonal subdivision surfaces, in: Shape Modeling International, 2002. Proceedings, IEEE. pp. 13–20.
- De Boor, C., Höllig, K., Riemenschneider, S., 1993. Box splines, volume 98 of Applied Mathematical Sciences. Springer-Verlag, New York.
- Dodgson, N.A., 2005. An heuristic analysis of the classification of bivariate subdivision schemes, in: Mathematics of Surfaces XI. Springer, pp. 161–183.
- Dodgson, N.A., Augsdörfer, U.H., Cashman, T.J., Sabin, M.A., 2009. Deriving box-spline subdivision schemes, in: Mathematics of Surfaces XIII, Springer. pp. 106–123.
- Donatelli, M., Novara, P., Romani, L., Serra-Capizzano, S., Sesana, D., 2019. A merged tuning of binary and ternary Loop’s subdivision. Computer Aided Geometric Design 69, 27–44.
- Doo, D., Sabin, M., 1978. Behaviour of recursive division surfaces near extraordinary points. Computer-Aided Design 10, 356–360.
- Du, Q., Faber, V., Gunzburger, M., 1999. Centroidal Voronoi tessellations: Applications and algorithms. SIAM Review 41, 637–676.
- Dyn, N., 2002. Analysis of convergence and smoothness by the formalism of laurent polynomials, in: Tutorials on Multiresolution in Geometric Modelling. Springer, pp. 51–68.
- Dyn, N., Levin, D., Gregory, J.A., 1990. A butterfly subdivision scheme for surface interpolation with tension control. ACM transactions on Graphics (TOG) 9, 160–169.
- Dyn, N., Levin, D., Liu, D., 1992. Interpolatory convexity-preserving subdivision schemes for curves and surfaces. Computer-Aided Design 24, 211–216.

- Dyn, N., Levin, D., Simoens, J., 2003. Face value subdivision schemes on triangulations by repeated averaging, in: *Curve and Surface Fitting: Saint Malo*, pp. 129–138.
- Farin, G., 1982. Designing C^1 surfaces consisting of triangular cubic patches. *Computer-Aided Design* 14, 253–256.
- Frederickson, P.O., 1971. Generalized Triangular Splines, Mathematics Report 7-71. Technical Report. Lakehead University.
- Jiang, C., Wang, J., Wallner, J., Pottmann, H., 2014. Freeform honeycomb structures. *Computer Graphics Forum* 33, 185–194.
- Kobbelt, L., 2000. $\sqrt{3}$ -subdivision, in: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co.. pp. 103–112.
- Kosinka, J., Sabin, M., Dodgson, N., 2013. Cubic subdivision schemes with double knots. *Computer Aided Geometric Design* 30, 45–57.
- Lacewell, D., Burley, B., 2007. Exact evaluation of Catmull-Clark subdivision surfaces near B-spline boundaries. *Journal of Graphics Tools* 12, 7–15.
- Lai, M.J., Schumaker, L.L., 2007. *Spline functions on triangulations*. Cambridge University Press.
- Loop, C., 1987. Smooth subdivision surfaces based on triangles. Master's thesis. University of Utah.
- Ma, W., 2005. Subdivision surfaces for CAD — an overview. *Computer-Aided Design* 37, 693–709.
- Oswald, P., Schröder, P., 2003. Composite primal/dual $\sqrt{3}$ subdivision schemes. *Computer Aided Geometric Design* 20, 135–164.
- Peters, J., 2014. Refinability of splines derived from regular tessellations. *Computer Aided Geometric Design* 31, 141–147.
- Peters, J., Reif, U., 1997. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics (TOG)* 16, 420–431.
- Peters, J., Reif, U., 2008. *Subdivision surfaces*. Springer.
- Peters, J., Wu, X., 2006. On the local linear independence of generalized subdivision functions. *SIAM Journal on Numerical Analysis* 44, 2389–2407.
- Pottmann, H., Eigensatz, M., Vaxman, A., Wallner, J., 2015. Architectural geometry. *Computers & Graphics* 47, 145–164.
- Prautzsch, H., 1984. Unterteilungsalgorithmen für multivariate Splines — ein geometrischer Zugang. Ph.D. thesis. Technische Universität Braunschweig.
- Prautzsch, H., Boehm, W., 2002. Box splines, in: Farin, G.E., Hoschek, J., Kim, M.S. (Eds.), *Handbook of Computer Aided Geometric Design*. Elsevier. chapter 10, pp. 255–283.
- Prautzsch, H., Umlauf, G., 1999. Triangular G^2 splines, in: *Curves and Surface Design*, pp. 335–342.
- Ramshaw, L., 1989. Blossoms are polar forms. *Computer Aided Geometric Design* 6, 323–358.
- Reif, U., 1998. Analyse und Konstruktion von Subdivisionsalgorithmen für Freiformflächen beliebiger Topologie. Habilitation. University of Stuttgart.
- Sabin, M.A., 1977. The use of piecewise forms for the numerical representation of shape. Ph.D. thesis. Hungarian Academy of Science, Budapest.
- Sabin, M.A., Bejancu, A., 2003. Boundary conditions for the 3-direction box-spline, in: *Mathematics of Surfaces*. Springer, pp. 244–261.
- Smith, J., Epps, D., Séquin, C., 2004. Exact evaluation of piecewise smooth Catmull-Clark surfaces using Jordan blocks. Technical Report. UC Berkeley.
- Stam, J., 1998a. Evaluation of Loop subdivision surfaces, in: *SIGGRAPH'98 CDROM Proceedings*.
- Stam, J., 1998b. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, in: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM. pp. 395–404.
- Velho, L., Zorin, D., 2001. 4–8 subdivision. *Computer Aided Geometric Design* 18, 397–427.
- Wang, W., Liu, Y., Yan, D., Chan, B., Ling, R., Sun, F., 2008. Hexagonal meshes with planar faces. Technical Report. The University of Hong Kong.
- Wang, Y., Yan, D.M., Liu, X., Tang, C., Guo, J., Zhang, X., Wonka, P., 2018. Isotropic surface remeshing without large and small angles. *IEEE Transactions on Visualization and Computer Graphics (Early Access)* .
- Warren, J., Weimer, H., 2001. *Subdivision methods for geometric design: A constructive approach*. Elsevier.
- Wei, X., Zhang, Y., Hughes, T.J., Scott, M.A., 2015. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Computer Methods in Applied Mechanics and Engineering* 291, 1–20.
- Zore, U., Jüttler, B., Kosinka, J., 2016. On the linear independence of truncated hierarchical generating systems. *Journal of Computational and Applied Mathematics* 306, 200–216.