Subdividing Barycentric Coordinates

Dmitry Anisimov · Chongyang Deng · Kai Hormann

Abstract

Barycentric coordinates are commonly used to represent a point inside a polygon as an affine combination of the polygon's vertices and to interpolate data given at these vertices. While unique for triangles, various generalizations to arbitrary simple polygons exist, each satisfying a different set of properties. Some of these generalized barycentric coordinates do not have a closed form and can only be approximated by piecewise linear functions. In this paper we show that subdivision can be used to refine these piecewise linear functions without losing the key barycentric properties. For a wide range of subdivision schemes, this generates a sequence of piecewise linear coordinates which converges to non-negative and C^1 continuous coordinates in the limit. The power of the described approach comes from the possibility of evaluating the C^1 limit coordinates and their derivatives directly. We support our theoretical results with several examples, where we use Loop or Catmull–Clark subdivision to generate C^1 coordinates, which inherit the favourable shape properties of harmonic coordinates or the small support of local barycentric coordinates.

Citation Info

Journal
Computer Aided Geometric Design

Volume
43, March 2016

Pages
172–185

Note
Proceedings of GMP

1 Introduction

Suppose we are given a planar n-sided simple polygon $\Omega \subset \mathbb{R}^2$ with $n \geq 3$ vertices $v_1, \ldots, v_n \in \mathbb{R}^2$. For any $p \in \mathbb{R}^2$, the values

$$\big[b_1(p), \ldots, b_n(p)\big] = b(p) \in \mathbb{R}^n$$

are called *barycentric coordinates* of p with respect to Ω, if

$$\sum_{i=1}^{n} b_i(p) = 1 \qquad \text{and} \qquad \sum_{i=1}^{n} b_i(p) v_i = p. \tag{1}$$

Non-negativity is sometimes mentioned as an additional condition [10], but since this precludes the existence of barycentric coordinates at points outside the convex hull of the vertices v_i, we prefer to consider the conditions in (1) as the *defining* properties and regard non-negativity as a *desirable* property only.

It is well known [27] that the barycentric coordinates of p are unique for $n = 3$, when Ω is a triangle, and they are non-negative if and only if $p \in \Omega$ in this case. Instead, for $n > 3$ the conditions in (1) describe an $(n-3)$-dimensional affine subspace of $\mathbb{R}^n$ from which $b(p)$ can be chosen. For example, Waldron [39] suggests to consider barycentric coordinates with minimal ℓ_2-norm and derives an explicit formula for computing them. He further shows that these *affine barycentric coordinates* are non-negative in a convex region that contains the barycentre $\bar{v} = (v_1 + \cdots + v_n)/n$ of Ω. Another example are Floater's *shape preserving coordinates* [7] which are well-defined and non-negative for any p in the kernel of Ω and have been used successfully for mesh parameterization [7] and morphing [9].

Both applications rely on *pointwise* barycentric coordinates, in the sense that $b(p)$ with the properties in (1) must be determined for a single point p in the kernel of some polygon Ω. Instead, other applications, like geometric modelling [22], colour interpolation [26], rendering [15], shape deformation [17], and image warping [40], require barycentric coordinates for all $p \in \Omega$ and consider $b(p)$ as a function of p over Ω. In this setting, the individual *barycentric coordinate functions* $b_i : \Omega \to \mathbb{R}$ must satisfy the *Lagrange property*

$$b_i(v_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases} \qquad i, j = 1, \ldots, n \tag{2}$$

in addition to the defining conditions in (1), so that the function $f : \Omega \to \mathbb{R}^d$ with

$$f(p) = \sum_{i=1}^{n} b_i(p) f_i \tag{3}$$

interpolates the data $f_1, \dots, f_n \in \mathbb{R}^d$ at the vertices $v_1, \dots, v_n$. Most applications further expect the barycentric coordinate functions to be *smooth*, so that the *barycentric interpolant* f in (3) is C^1 or even C^2 continuous. And for some applications it is crucial that the coordinates are *non-negative*, because this guarantees that the interpolated values $f(p)$ are contained in the convex hull of the data.

1.1 Related work

Wachspress [38] was the first to describe a construction of rational barycentric coordinate functions for *convex* polygons in the context of generalized finite element methods, but these *Wachspress coordinates* are not well-defined for arbitrary simple polygons. The same holds for *discrete harmonic coordinates*, which arise from the classical piecewise linear finite element approximation to Laplace's equation [36] and have been applied for computing discrete minimal surfaces [29] and mesh parameterization [6]. *Mean value coordinates* [8] overcome this drawback, as they are well-defined even for sets of nested simple polygons and for any $p \in \mathbb{R}^2$ [13]. However, mean value coordinates can be negative inside concave polygons, and the same is true for *metric* [37], *moving least squares* [24], *Poisson* [18], and *cubic mean value coordinates* [19]. Positivity inside arbitrary simple polygons is guaranteed by *positive mean value* [20] and *positive Gordon–Wixom coordinates* [23], but both constructions deliver only C^0 continuous coordinate functions.

All the aforementioned constructions provide *closed-form coordinates*, which can be evaluated exactly for any $p \in \Omega$ in a finite number of steps. At the same time, neither of these coordinates are smooth *and* positive inside non-convex polygons. So far the only barycentric coordinates known to have both properties are the *harmonic* [16], *maximum entropy* [14], and *local barycentric coordinates* [42], but they all are *computational coordinates* in the sense that they lack a closed-form expression and must be treated numerically.

For example, harmonic coordinates can be approximated by using the *complex variable boundary element method* [41, Sec. 6.1] or the *method of fundamental solutions* [25, Sec. 5]. The advantage of both approaches is that the resulting coordinates are smooth and harmonic and can be written in closed form after initially solving a rather small but dense linear system, but they only approximate the piecewise linear boundary conditions and thus do not satisfy the Lagrange property.

Another common strategy for computing harmonic coordinates [6, 16] is first to create a dense triangulation of Ω, then to fix the barycentric coordinates of the boundary vertices according to the Lagrange property (2) and such that the coordinates are linear along the edges of Ω, and finally to determine the coordinates at the interior vertices using the standard finite element discretization of the Laplace equation with Dirichlet boundary conditions. This approach is quite efficient, because it only requires solving a sparse linear system, but the resulting coordinate functions are merely *piecewise linear* approximations of the true harmonic coordinates. Local barycentric coordinates are approximated similarly, except that computing the coordinates at the interior vertices is more involved as it leads to a convex optimization problem with a non-smooth target function [42, Sec. 4]. However, the advantage of the resulting coordinate functions is that their support is smaller than the support of harmonic coordinate functions.

In both cases a *global* problem is solved to determine the barycentric coordinates for all interior vertices simultaneously. In contrast, maximum entropy coordinates are computed for any $p \in \Omega$ by solving a *local* convex optimization problem, which in turn can be done very efficiently with Newton's method [14, Sec. 5].

1.2 Contributions

In this paper we describe a novel way to construct non-negative barycentric coordinate functions. The main idea is to start with a piecewise linear approximation of harmonic or local barycentric coordinates over a coarse triangulation of Ω and then to use *subdivision* [44] to refine the coordinate functions (Section 2). While the coordinate functions remain piecewise linear after any finite number of subdivision steps, the refinement process gives C^1 continuous and non-negative coordinates in the limit for many common subdivision schemes, and these limit coordinates can be evaluated like maximum entropy coordinates by solving a local convex optimization problem (Section 2.1). In particular, we focus on *Loop subdivision* [21] and show that the resulting C^1 limit coordinate functions combine the favourable shape properties of harmonic coordinates or the small support of local barycentric coordinates with the possibility of evaluating them and their derivatives efficiently at any $p \in \Omega$ (Section 3). We further discuss briefly how to obtain similar results with *Catmull–Clark subdivision* [3] (Section 4). Our main contributions are:

- We prove that subdividing piecewise linear barycentric coordinates keeps all the desired properties, even in the limit, as long as the subdivision rules respect certain conditions.

- We show for Loop and Catmull–Clark subdivision how to avoid fold-overs at concave corners of Ω which would otherwise lead to not well-defined limit coordinate functions.
- We present several examples that illustrate the properties and advantages of the proposed approach.

2 Refining piecewise linear barycentric coordinates

Our main observation, which motivated us to investigate the idea of subdividing barycentric coordinates, is that affine combinations of points and barycentric coordinates commute in the following sense.

Lemma 1. *Suppose we are given m points $p_1,\dots,p_m \in \mathbb{R}^2$ with barycentric coordinates $b(p_1),\dots,b(p_m)$ and some weights $\alpha_1,\dots,\alpha_m \in \mathbb{R}$ which sum to one, $\sum_{j=1}^m \alpha_j = 1$. Let $p = \sum_{j=1}^m \alpha_j p_j \in \mathbb{R}^2$ be the point given by the affine combination of the points p_j with the weights α_j and $b(p) = \sum_{j=1}^m \alpha_j b(p_j)$ be the affine combination of the coordinates $b(p_j)$ with the same weights. Then $b(p)$ are barycentric coordinates of p. Moreover, if the coordinates $b(p_j)$ and the weights α_j are non-negative, then so are the coordinates $b(p)$.*

Proof. To prove the first statement, we refer to (1) and observe that

$$\sum_{i=1}^n b_i(p) = \sum_{i=1}^n \sum_{j=1}^m \alpha_j b_i(p_j) = \sum_{j=1}^m \alpha_j \sum_{i=1}^n b_i(p_j) = \sum_{j=1}^m \alpha_j = 1$$

and

$$\sum_{i=1}^n b_i(p) v_i = \sum_{i=1}^n \sum_{j=1}^m \alpha_j b_i(p_j) v_i = \sum_{j=1}^m \alpha_j \sum_{i=1}^n b_i(p_j) v_i = \sum_{j=1}^m \alpha_j p_j = p.$$

The second statement follows, because convex combinations of non-negative values are non-negative. □

Another well known fact, which turns out to be useful in this context, is that affine combinations in general, and in particular those of barycentric coordinates, commute with linear functions.

Lemma 2. *If the barycentric coordinates $b(p_j)$ of the points p_j in Lemma 1 lie on a common linear function, that is, $b(p_j) = Ap_j + c$ for some $A \in \mathbb{R}^{n\times 2}$, $c \in \mathbb{R}^n$, and $j = 1,\dots,n$, then so does their affine combination $b(p)$.*

Proof. The statement holds because

$$b(p) = \sum_{j=1}^m \alpha_j b(p_j) = \sum_{j=1}^m \alpha_j (Ap_j + c) = A\left(\sum_{j=1}^m \alpha_j p_j\right) + c\left(\sum_{j=1}^m \alpha_j\right) = Ap + c.$$

□

Suppose now that $\mathcal{T}^0$ is a triangulation of Ω and that we are given for each vertex p of $\mathcal{T}^0$ some initial barycentric coordinates $b(p)$. We then consider the piecewise linear function $b^0 = [b_1^0,\dots,b_n^0]\colon \mathcal{T}^0 \to \mathbb{R}^n$ which interpolates the given barycentric coordinates at the vertices of $\mathcal{T}^0$.

Corollary 3. *If the initial barycentric coordinates at the vertices v_i of Ω are*

$$b(v_i) = \boldsymbol{\delta}_i = [\delta_{1,i},\dots,\delta_{n,i}], \qquad i = 1,\dots,n, \tag{4}$$

then the components b_i^0 of b^0 are barycentric coordinate functions. If all initial barycentric coordinates are non-negative, then so are the functions b_i^0.

Proof. For any $p \in \Omega$, let $T = [p_1, p_2, p_3]$ be the triangle in $\mathcal{T}^0$ that contains p, so that $p = \sum_{j=1}^3 \alpha_j p_j$, where α_j are the unique barycentric coordinates of p with respect to T. By the definition of b^0 we have $b^0(p) = \sum_{j=1}^3 \alpha_j b(p_j)$, and Lemma 1 assures not only that $b^0(p)$ are valid barycentric coordinates of p in the sense of (1), but also the statement about non-negativity. Condition (4) further guarantees that b_i^0 satisfies the Lagrange property (2). □

We then refine $\mathcal{T}^0$ successively with some linear subdivision scheme $\mathcal{S}$ [44] to generate the sequence of triangulations $\mathcal{T}^0, \mathcal{T}^1, \dots$ and apply the subdivision rules not only to the (x, y) coordinates of the vertices, but also to the associated barycentric coordinates. That is, if the vertex p of $\mathcal{T}^{k+1}$ is generated by the affine combination $p = \sum_{j=1}^m \alpha_j p_j$ of some vertices $p_1,\dots,p_m$ of $\mathcal{T}^k$, then we associate with p the values $b(p) = \sum_{j=1}^m \alpha_j b(p_j)$, and it follows from Lemma 1 that $b(p)$ are valid barycentric coordinates of p. As above, we now consider at each level k the piecewise linear function $b^k = [b_1^k,\dots,b_n^k]\colon \mathcal{T}^k \to \mathbb{R}^n$ which interpolates the generated barycentric coordinates at the vertices of $\mathcal{T}^k$ (see Figure 1).

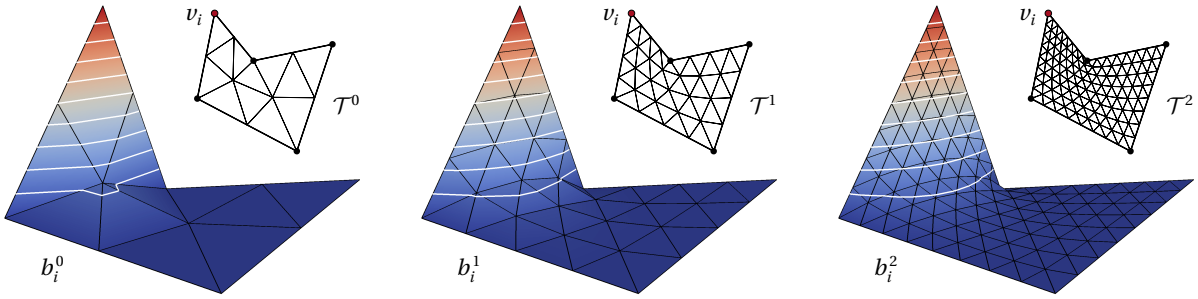


Figure 1: Main idea of refining piecewise linear barycentric coordinates: A triangulation $\mathcal{T}^0$ of the polygon Ω is refined by a linear subdivision scheme with special rules to keep the boundary fixed. In parallel, the same subdivision rules are applied to the barycentric coordinates associated with the vertices of the triangulation, thus creating a sequence of piecewise linear barycentric coordinate functions b_i^k (shown for the red vertex) with a C^1 continuous limit. The white curves are the contour lines at $0.1, 0.2, \ldots, 0.9$.

Theorem 4. *Let $\mathcal{S}$ be a subdivision scheme that*

1. *is convergent,*
2. *generates C^1 continuous limits,*
3. *is equipped with boundary rules for interpolating corner vertices and preserving straight boundary segments.*

Further assume that the triangulations $\mathcal{T}^k$ are regular in the sense that they do not contain any degenerate or flipped triangles, even in the limit. Then the components b_i^k of b^k converge to C^1 continuous barycentric coordinate functions $b_i^\infty : \Omega \to \mathbb{R}$ as $k \to \infty$. Moreover, if the initial barycentric coordinates at the vertices of $\mathcal{T}^0$ and the weights of the subdivision rules are non-negative, then so are the b_i^∞.

Proof. Using the appropriate boundary rules along the edges of Ω ensures that $\mathcal{T}^k$ is a triangulation of Ω. Moreover, tagging the vertices of Ω as corners and applying to them the interpolating subdivision rule guarantees that condition (4) is preserved at any level k. With the same reasoning as in the proof of Corollary 3 we then conclude that the b_i^k are barycentric coordinate functions. Note that we have to assume here that $\mathcal{T}^k$ is regular, because otherwise it could happen that some $p \in \Omega$ is contained in more than one triangle of $\mathcal{T}^k$ and then b_i^k would not be well-defined.

To study the limit behaviour of this subdivision process, we recall that the *natural parameterization* of a subdivision surface is the one with respect to the midpoint-subdivided control mesh [44]. In our setting, this means that we use Ω as our domain, consider the sequence of triangulations $\mathcal{D}^0, \mathcal{D}^1, \ldots$, where $\mathcal{D}^0 = \mathcal{T}^0$ and $\mathcal{D}^{k+1}$ is derived from $\mathcal{D}^k$ by midpoint subdivision, and regard $\mathcal{T}^k$ as the image of the piecewise linear function $v^k : \Omega \to \Omega$ that maps from each triangle in $\mathcal{D}^k$ to the corresponding triangle in $\mathcal{T}^k$ (see Figure 2). Under the given conditions on $\mathcal{S}$, this sequence of functions converges to a C^1 continuous mapping $\boldsymbol{v} : \Omega \to \Omega$. Likewise, subdividing the initial barycentric coordinates gives a C^1 continuous mapping $\boldsymbol{b} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n] : \Omega \to \mathbb{R}^n$ in the limit. Putting both together, we conclude that the barycentric coordinate functions b_i^k converge to the C^1 continuous functions $b_i^\infty = \boldsymbol{b}_i \circ \boldsymbol{v}^{-1}$. Note that we have to assume here the regularity of $\mathcal{T}^k$ in the limit in order to ensure that $\boldsymbol{v}^{-1}$ exists and is C^1 continuous, according to the inverse function theorem. The functions b_i^∞ are barycentric coordinate functions, because

$$\sum_{i=1}^{n} b_i^\infty(p) = \sum_{i=1}^{n} \lim_{k\to\infty} b_i^k(p) = \lim_{k\to\infty} \sum_{i=1}^{n} b_i^k(p) = \lim_{k\to\infty} 1 = 1$$

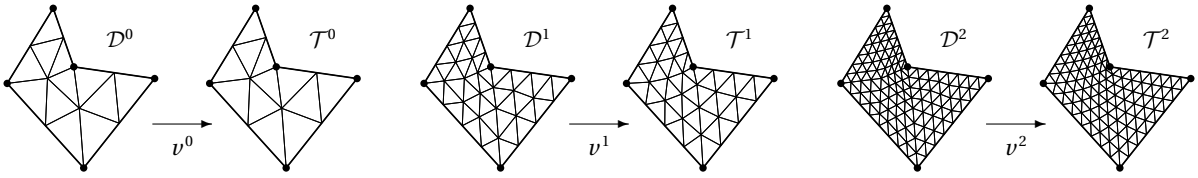


Figure 2: Natural parameterization v^k of the subdivided triangulation $\mathcal{T}^k$ over the refined domain $\mathcal{D}^k$ for $k = 0, 1, 2$.

and

$$\sum_{i=1}^{n} b_i^\infty(p)v_i = \sum_{i=1}^{n} \lim_{k\to\infty} b_i^k(p)v_i = \lim_{k\to\infty} \sum_{i=1}^{n} b_i^k(p)v_i = \lim_{k\to\infty} p = p,$$

and the statement about the non-negativity of b_i^∞ follows immediately from the given conditions. □

The conditions on $\mathcal{S}$ in Theorem 4 are not very restrictive and satisfied by many popular subdivision schemes [44, 2]. However, we recommend to use *approximating* schemes, because *interpolating* schemes, like the *butterfly scheme* [5, 45] have subdivision rules with negative coefficients, so that the non-negativity of the limit coordinates b^∞ is not guaranteed. We further note that Corollary 3 and Theorem 4 work for any initial data, but in our examples we mainly focus on the setting where the initial piecewise linear barycentric coordinates b^0 are either harmonic or local barycentric coordinates, computed for some triangulation $\mathcal{T}^0$ of Ω. By construction, the initial coordinate functions b_i^0 are linear along the edges of Ω in these cases, and it follows from Lemma 2 and the third condition on $\mathcal{S}$ in Theorem 4 that the same is true for b_i^k and the limit coordinate functions b_i^∞.

2.1 Evaluation

For the evaluation of the limit coordinates b^∞, there are three possible scenarios. First, there are many applications, where it is sufficient to have a piecewise linear approximation of the coordinates. In this situation, we simply carry out a finite number of, say $k = 5$ or $k = 6$ subdivision steps, and take b^k as the desired piecewise linear approximation over $\mathcal{T}^k$. We can further use the limit rules of $\mathcal{S}$ to snap the vertices p of $\mathcal{T}^k$ to their limit positions $\bar{p}$, thus giving a new triangulation $\bar{\mathcal{T}}^k$. Concurrently we apply the same limit rules to the corresponding coordinates $b^k(p)$ to compute $b^\infty(\bar{p})$. Overall this results in piecewise linear coordinates $\bar{b}^k$ over $\bar{\mathcal{T}}^k$, which *interpolate* the limit coordinates at the vertices $\bar{p}$ of $\bar{\mathcal{T}}^k$ instead of only *approximating* them.

The other two scenarios require the availability of a general routine for evaluating the limit surfaces generated by $\mathcal{S}$ at arbitrary parameter values, which in our setting allows to compute $\boldsymbol{v}(p)$ and $\boldsymbol{b}(p)$ at any $p \in \Omega$. For spline subdivision schemes which generate polynomial patches in regular regions, such a routine with constant time complexity can be designed by following the ideas of Stam [34, 35], and non-polynomial schemes can be evaluated with the approach of Schaefer and Warren [31, 32]. On the one hand, we can then map any $p \in \Omega$ to its limit position $\bar{p} = \boldsymbol{v}(p) \in \Omega$ and compute the limit coordinates $b^\infty(\bar{p}) = \boldsymbol{b}(p)$ of $\bar{p}$. This is sufficient, for example, for applications which require to evaluate b^∞ at a dense set of points, but where the exact positions of the points do not matter. On the other hand, we can also determine the limit coordinates $b^\infty(p)$ of $p \in \Omega$ itself by first finding $q = \boldsymbol{v}^{-1}(p)$, which in turn requires solving the local convex optimization problem

$$\min_{q\in\Omega} \|p - \boldsymbol{v}(q)\|^2, \tag{5}$$

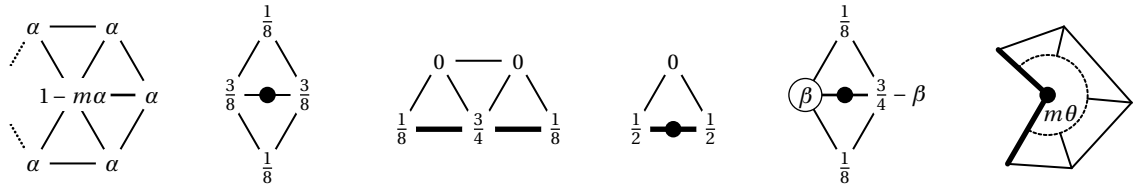
for example with Newton's method [28]. Once q is found we compute the limit coordinates of p as $b^\infty(p) = \boldsymbol{b}(q)$.

2.2 Connection to standard surface subdivision

Before continuing with some concrete examples, we would like to point out a different perspective on the subdivision process described above. Suppose we attach the i-th barycentric coordinate $b_i(p)$ as a z-coordinate to each vertex p of the triangulation $\mathcal{T}^k$. This turns $\mathcal{T}^k$ into the 3D triangle mesh $\mathcal{M}_i^k$, which is nothing but the *graph* of the barycentric coordinate function b_i^k, and generating $\mathcal{M}_i^{k+1}$ from $\mathcal{M}_i^k$ is just the standard surface subdivision process. Under the given conditions on $\mathcal{S}$ in Theorem 4, it is then clear that the sequence of meshes $\mathcal{M}_i^0, \mathcal{M}_i^1, \ldots$ converges to a C^1 continuous limit surface $\mathcal{M}_i^\infty$, and the regularity of $\mathcal{T}^k$ in the limit guarantees that $\mathcal{M}_i^\infty$ is the graph of a function, namely the limit coordinate function b_i^∞.

3 Loop coordinates

In order to verify the theoretical results from the previous section we decided to use Loop subdivision [21] with the modification proposed by Biermann et al. [1]. That is, we mark vertices and edges of Ω as corners and creases to preserve the boundary of the polygon and use the subdivision rules in Figure 3, where the parameter α for an interior vertex with valency m is $\alpha = \left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{m}\right)^2\right)/m$. The standard rules are used



(a) (b) (c) (d) (e) (f)

Figure 3: Standard Loop subdivision rules for interior vertices (a), interior edges (b), boundary vertices (c), and boundary edges (d). Corner vertices are simply interpolated. The modified edge rule (e) depends on the interior angle at the adjacent corner vertex (f).

everywhere, except at interior edges adjacent to exactly one corner. For these edges, the corner is weighted by the modified coefficient $\beta = (1+\cos\theta)/4$, where $m\theta$ is the interior angle at the corner and m is the number of adjacent triangles.

As the subdivision rules have non-negative weights, they generate non-negative coordinate functions b_i^∞ in the limit. These rules further guarantee that the coordinate functions are C^2 almost everywhere in the interior and along the edges of Ω, and they are C^1 at extraordinary interior vertices with valency other than 6 and at convex corners. The b_i^∞ are only C^0 at concave corners, but this is not surprising, because non-negative coordinate functions cannot be C^1 at such corners (see Figure 4).

3.1 Evaluation

To evaluate these *Loop coordinates* b^∞, we implemented the three strategies outlined in Section 2.1 in *C++* on a MacBook Pro with 2.4 GHz Intel Core i7 processor and 8 GB RAM. The first option is to subdivide the triangulation $\mathcal{T}^0$ and the initial barycentric coordinates b^0 until $\mathcal{T}^k$ has about one million vertices and to snap both the vertices p of $\mathcal{T}^k$ and the corresponding coordinates $b^k(p)$ to the limit using the usual limit rules [21]. This gives a rather detailed piecewise linear interpolant of b^∞. Our implementation takes about 2 seconds for subdividing the (x,y) coordinates of the vertices of the triangulation and managing the data structures, plus another $0.02n$ seconds for subdividing the associated barycentric coordinates, where n is the number of vertices of Ω. By means of the tangent vector rules, we can even determine the gradients ∇b_i^∞ of the limit coordinate functions at the limit points at an additional cost of $0.2+0.05n$ seconds. Note that computing harmonic coordinates for a triangulation with the same number of vertices costs about $0.2n$ seconds in our implementation, which is based on *Eigen* [12], plus 15 seconds for assembling and factorizing the matrix. Hence, it is even a bit faster for small n, but the resulting piecewise linear coordinate functions do not interpolate the true harmonic coordinates at the vertices and gradients can only be approximated.

The second option is to evaluate for any $p \in \Omega$ the limit mappings $\boldsymbol{v}$ and $\boldsymbol{b}$, so as to get the limit coordinates $b^\infty(\bar{p}) = \boldsymbol{b}(p)$ at $\bar{p} = \boldsymbol{v}(p)$. To this end, we first subdivide $\mathcal{T}^0$ and b^0 twice in a preprocessing step, to separate extraordinary vertices, and then find the triangle T in $\mathcal{T}^2$ that contains p. If T is not adjacent to the boundary of $\mathcal{T}^2$, then we use Stam's algorithm [35], otherwise we resort to the method of Zorin and Kristjansson [43], which is slightly more complex but works for points near the boundary. With our implementation, which uses a quadtree to store the triangles of $\mathcal{T}^2$, evaluating one million points this way takes about 3.2 seconds for finding the triangle T and computing $\bar{p}$, plus $0.1n$ seconds for evaluating $b^\infty(\bar{p})$. We can further use Stam's approach to compute first derivatives of b_i^∞ at $\bar{p}$ at roughly the same cost and even second derivatives, except

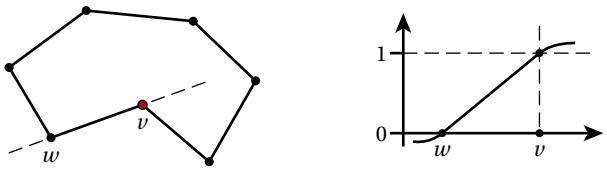


Figure 4: Consider the barycentric coordinate function for the concave vertex v (red) and the cross section along the line defined by this vertex and the neighbouring vertex w (dashed line). If this function is C^1 at v, then it must be greater than one in the interior of Ω, which implies that another coordinate function is negative.

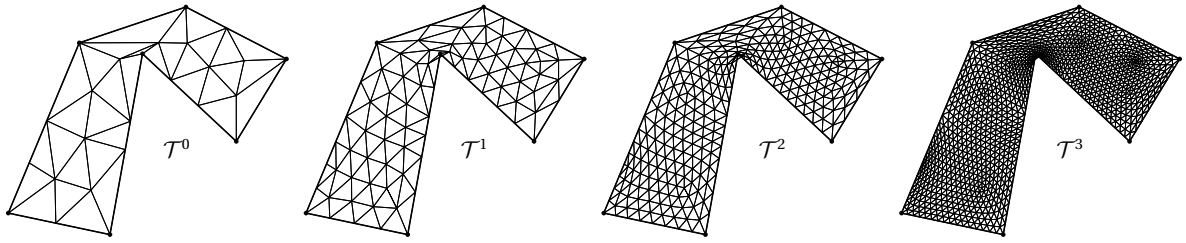


Figure 5: Example of an initial triangulation $\mathcal{T}^0$ for which the subdivided triangulations $\mathcal{T}^k$ fold over at the concave corner.

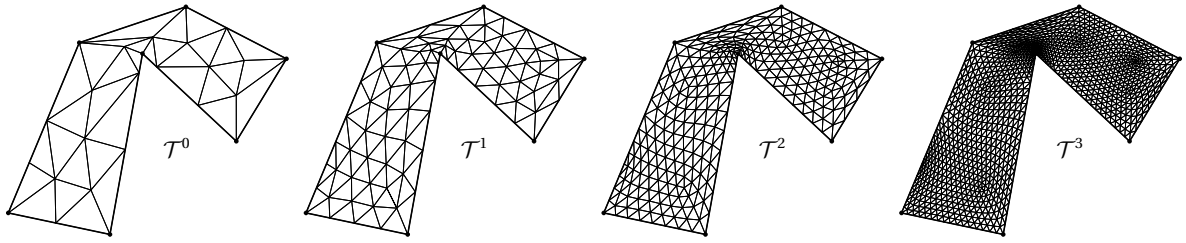


Figure 6: The foldovers in Figure 5 can be avoided by our vertex adjustment strategy (see Figure 7).

when p is an extraordinary vertex of $\mathcal{T}^2$ without well-defined second derivatives. To compute derivatives at points near the boundary, we subdivide the triangulation around p locally until the triangle that contains p is not adjacent to the boundary anymore before calling Stam's routine, because Zorin and Kristjansson do not discuss how to compute derivatives with their method. However, the cost of these local subdivisions has a negligible effect on the average runtime.

The third option is to compute $b^\infty(p)$ for any $p \in \Omega$, which requires to solve the optimization problem (5). We implemented a simple Newton method with adaptive step size, taking advantage of the fact that we can use Stam's method as explained above to get the gradient and the Hessian of the objective function. At extraordinary vertices, where the Hessian is undefined, we resort to a finite difference approximation of the Hessian. Our experiments show that the optimal point $q = \boldsymbol{\nu}^{-1}(p)$ is usually found in less than three iterations with an accuracy of 10^{-7} at an average cost of $2 \cdot 10^{-6}$ seconds per point. Note that this cost does not depend on n, since it is a problem in $\mathbb{R}^2$. Once q is found, we proceed to compute $b^\infty(p) = \boldsymbol{b}(q)$ as in the second option above. Overall, our implementation takes about $5 + 0.1n$ seconds for evaluating one million points this way. This is roughly on par with the runtime of our implementation of maximum entropy coordinates, which takes about $2 + 0.15n$ seconds for the same task.

While the third option is the least efficient, it is the only one that delivers the limit coordinates $b^\infty(p)$ at an arbitrary $p \in \Omega$. Moreover, the additional cost with respect to the second option becomes marginal for large n, and in comparison to the first option it requires less memory, as it needs to store only $\mathcal{T}^2$ instead of $\mathcal{T}^k$.

3.2 Concave corners

In the reasoning above we tacitly assumed that the subdivision process gives regular triangulations $\mathcal{T}^k$, even in the limit as $k \to \infty$. However, as noticed by Biermann et al. [1], foldovers may occur at concave corners, not only in the limit, but already after a small number of subdivision steps (see Figure 5). Consequently, the limit coordinates will not be well-defined in these regions. However, we can avoid this problem (see Figure 6) by modifying $\mathcal{T}^0$ *before* determining the initial barycentric coordinates b^0.

To this end, we adjust the positions of the vertices p_i in the one-ring neighbourhood of a concave corner p as shown in Figure 7. That is, we first determine the length

$$r = \min\{r_0, r_1, \ldots, r_m\},$$

where $r_i = \|p - p_i\|$, of the shortest edge adjacent to the concave corner. We then place all neighbours regularly spaced on a circle with radius r around p,

$$p_0' = p + (p_0 - p)r/r_0, \qquad p_i' = p + R_{i\theta}(p_0' - p), \qquad i = 1, \ldots, m, \tag{6}$$

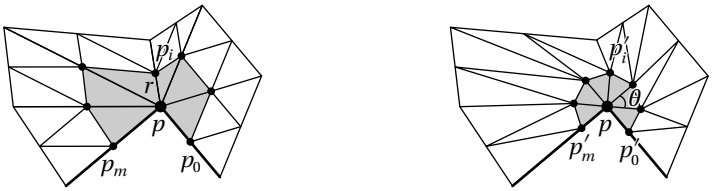


Figure 7: Our vertex adjustment strategy relocates the vertices in the one-ring neighbourhood of a concave corner (left) so that all adjacent triangles have the same shape and size (right).

where θ is defined as above and R_γ denotes the rotation matrix for counterclockwise rotation by γ. If this vertex adjustment strategy creates foldovers of $\mathcal{T}^0$ in the 2-ring neighbourhood of p, then we repeatedly halve r until the foldovers disappear. Note that this strategy generally requires that the one-ring neighbourhoods of the concave corners do not contain common vertices.

Theorem 5. *If the neighbours of a concave corner p have been adjusted with the strategy in* (6)*, then the triangulations $\mathcal{T}^k$ are regular around p, even in the limit.*

Proof. We first note that the adjusted neighbours of p satisfy

$$p_{i\pm1} = p + R_{\pm\theta}(p_i - p), \qquad i = 1,\dots,m-1.$$

Recalling that

$$R_\theta + R_{-\theta} = 2\cos\theta\, I = (8\beta - 2)I,$$

where β is defined as above and I denotes the identity matrix, we find that after one subdivision step with the modified edge rule in Figure 3 (e), the new interior neighbours $\tilde{p}_i$ of p for $i = 1,\dots,m-1$ are just the edge midpoints of the old interior edges,

$$\begin{aligned}
\tilde{p}_i &= [8\beta p + (6-8\beta)p_i + p_{i-1} + p_{i+1}]/8 \\
&= [8\beta p + (6-8\beta)p_i + 2p + (R_\theta + R_{-\theta})(p_i - p)]/8 \\
&= [8\beta p + (6-8\beta)p_i + 2p + (8\beta - 2)(p_i - p)]/8 \\
&= [p + p_i]/2.
\end{aligned}$$

As the same holds for the new boundary neighbours $\tilde{p}_0$ and $\tilde{p}_m$, due to the boundary edge rule in Figure 3 (d), we conclude that each subdivision step simply scales the one-ring neighbourhood of p by a factor of $1/2$. This clearly avoids foldovers at p, even in the limit. □

3.3 Internal foldovers

While our vertex adjustment strategy takes care of foldovers at concave corners, interior foldovers may still occur in the interior of $\mathcal{T}^k$, even for convex initial triangulations $\mathcal{T}^0$ where all interior vertices are regular with valency 6 (see Figure 8). A formal analysis of this problem is beyond the scope of this paper, but we observed that this problem does not appear if we construct the initial triangulation as shown in Figure 9.

Given a polygon Ω and a target edge length h, we first sample each edge of Ω with uniformly spaced vertices such that the spacing is as close as possible to h. We then use *Triangle* [33] to compute a *conforming constrained Delaunay triangulation* of Ω which contains the sample vertices, does not create any further

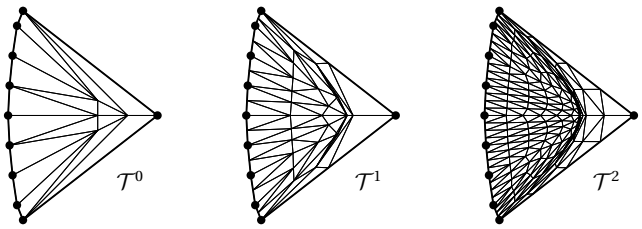


Figure 8: For this convex triangulation with regular interior vertices, foldovers occur in the interior after two subdivision steps.

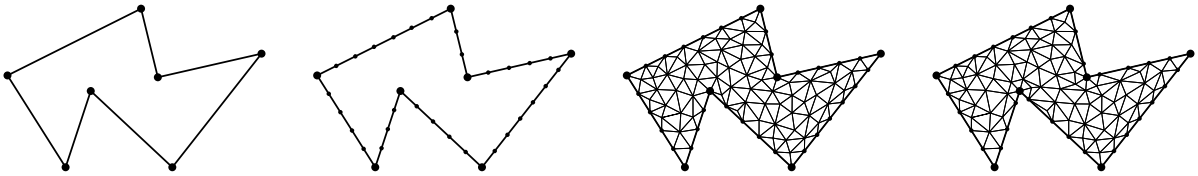


Figure 9: To create the initial triangulation $\mathcal{T}^0$ for a given polygon Ω (a), we first specify uniformly spaced vertices on the edges of Ω (b), then compute a constrained Delaunay triangulation (c), and finally modify the one-ring neighbourhood of each concave corner with our vertex adjustment strategy (d).

boundary vertices, and has triangles with areas less than $A = h^2\sqrt{3}/4$, the area of the equilateral triangle with target edge length h. This usually generates a triangulation with a maximum edge length $h^\star$ close to h and not too many extraordinary vertices. In a final step we apply the vertex adjustment strategy from Section 3.2 to create the initial triangulation $\mathcal{T}^0$ of Ω.

To test our conjecture that the limit mapping $\boldsymbol{\nu}$ is regular so that $\boldsymbol{\nu}^{-1}$ exists and the limit coordinates are well-defined, we generated 100 random simple polygons and triangulated them for different values of h. We then subdivided each initial triangulation until the number of triangles was above one million and applied three tests. For each triangle with three regular vertices (usually more than 99.9% of all triangles), we checked the condition in [11, Lemma 3] to verify that the corresponding limit patch is regular. We further computed the limit tangents at all extraordinary vertices and checked that the limit mapping does not fold at these vertices. Both tests restrict the potential occurrence of foldovers to the triangles adjacent to extraordinary vertices and we evaluated the limit tangents at 1000 random points inside each of these triangles as explained in Section 3.1. All initial test triangulations passed these tests, which makes us confident that our conjecture is true.

3.4 Examples

Figure 10 shows some examples of Loop coordinate functions for harmonic initial coordinates. Despite the low resolution of the triangulation $\mathcal{T}^0$, the functions are smooth and no visual artefacts are recognizable at the extraordinary interior vertices. Not too surprisingly, they actually look very similar to harmonic coordinates, and Figures 11 and 12 further illustrate this behaviour. In both examples, we first computed harmonic coordinates over a mesh with two million triangles and took this approximation as referential true harmonic coordinates b^H. As expected, the log-log plots show that the piecewise linear harmonic coordinate functions b_i^0 over $\mathcal{T}^0$ converge to b_i^H as the maximum edge length $h^\star$ of $\mathcal{T}^0$ tends to 0, and that the same holds for the *Dirichlet energy*

$$D(f) = \frac{1}{2}\int_\Omega \|\nabla f\|^2, \qquad f : \Omega \to \mathbb{R},$$

which is of course minimal for b_i^H. The plots also show that the Loop coordinate functions b_i^∞ with b_i^0 as initial coordinates and their Dirichlet energies converge at the same rate and are consistently closer to b_i^H. The behaviour is confirmed by the error visualizations which illustrate that Loop subdivision effectively smoothes out the error between approximate and true harmonic coordinates.

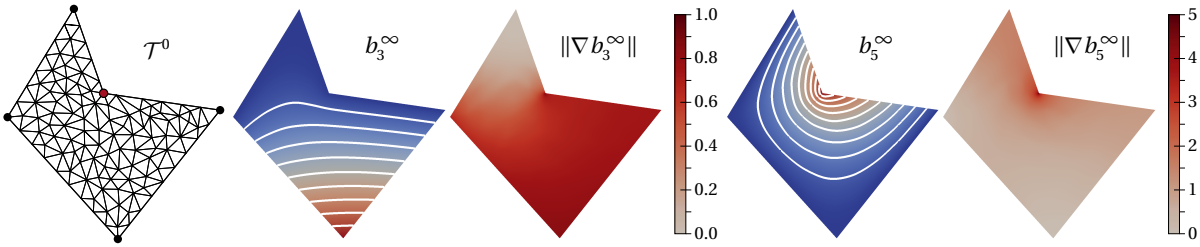


Figure 10: Example of Loop coordinate functions and the norm of their gradients (shown for the red vertices) using piecewise linear harmonic coordinates over $\mathcal{T}^0$ as initial coordinates b^0.

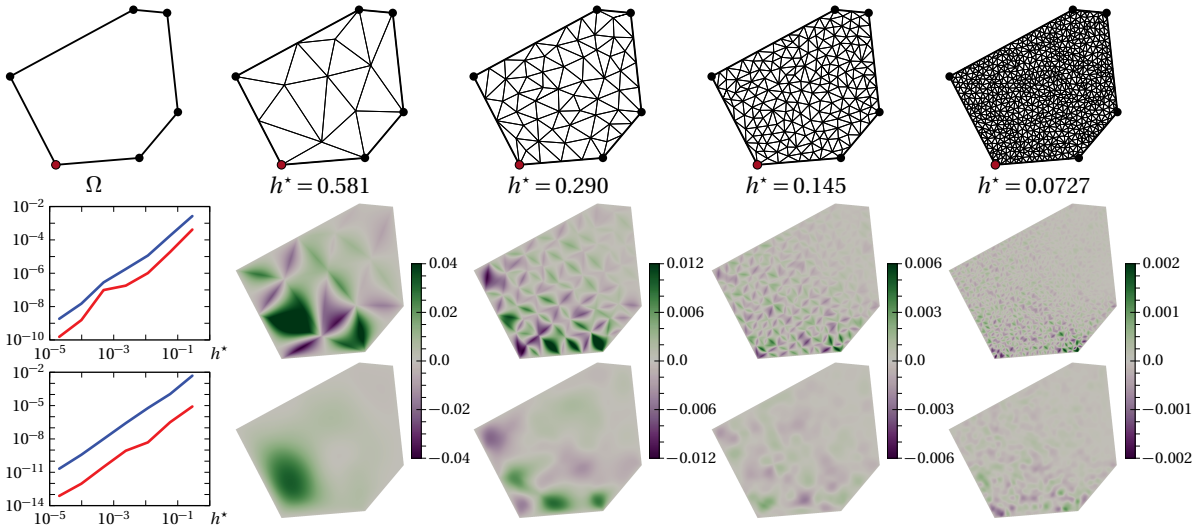


Figure 11: Comparison of the errors $b_i^H - b_i^0$ (centre) and $b_i^H - b_i^\infty$ (bottom) between the true harmonic coordinate function b_i^H, the piecewise linear approximation b_i^0, and the Loop coordinate function b_i^∞ for different resolutions of the triangulation $\mathcal{T}^0$ with maximum edge lengths $h^\star$. The top log-log plot shows the maximum errors $\|b_i^H - b_i^0\|_\infty$ (blue) and $\|b_i^H - b_i^\infty\|_\infty$ (red) over $h^\star$, and the bottom log-log plot shows the differences of the Dirichlet energies $D(b_i^0) - D(b_i^H)$ (blue) and $D(b_i^\infty) - D(b_i^H)$ (red).

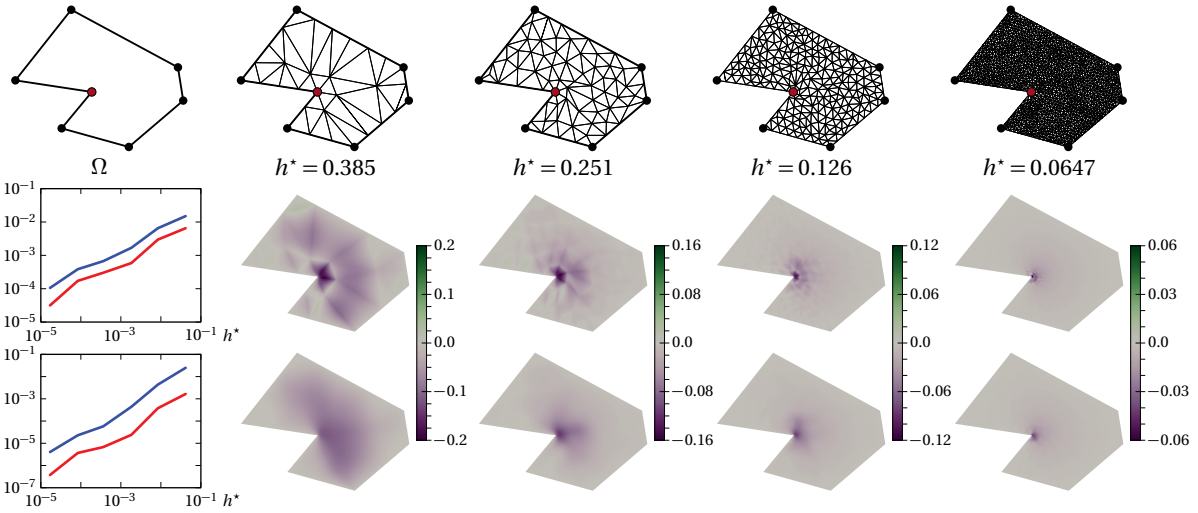


Figure 12: Comparison of the errors $b_i^H - b_i^0$ (centre) and $b_i^H - b_i^\infty$ (bottom) between the true harmonic coordinate function b_i^H, the piecewise linear approximation b_i^0, and the Loop coordinate function b_i^∞ for different resolutions of the triangulation $\mathcal{T}^0$ with maximum edge lengths $h^\star$. The top log-log plot shows the maximum errors $\|b_i^H - b_i^0\|_\infty$ (blue) and $\|b_i^H - b_i^\infty\|_\infty$ (red) over $h^\star$, and the bottom log-log plot shows the differences of the Dirichlet energies $D(b_i^0) - D(b_i^H)$ (blue) and $D(b_i^\infty) - D(b_i^H)$ (red).

For the example in Figure 13, the authors of [42] provided us with local barycentric coordinates for different resolutions of $\mathcal{T}^0$. Although the theory suggests that these coordinate functions are locally supported, the numerical solver used in [42] generates small function values even outside the probable support and Zheng et al. suggest to consider all values below 10^{-4} as numerically zero. We modified their data in the following way. For each vertex p of $\mathcal{T}^0$ with one or more coordinates $b_i(p) < 10^{-4}$ we set $b_i(p)$ to exact zero and perturbed the other coordinates in a least squares sense to restore the barycentric properties in (1). We then used these modified coordinates as b^0. The plots show that the corresponding Loop coordinates are truly locally supported and that the support is slightly larger, but also smoother than the numerical support of the original local coordinates. We further observe that the shape of the Loop coordinates for the initial triangulations with 1784 and 6979 vertices is visually the same, which suggests that, given the exponential

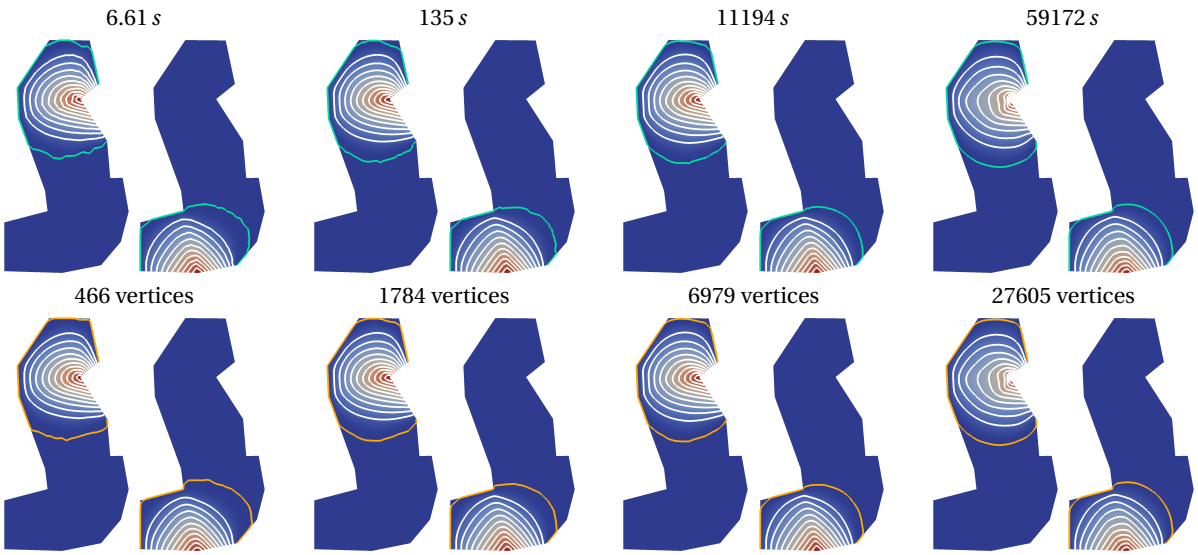


Figure 13: Comparison of local barycentric coordinates (top) and corresponding Loop coordinates (bottom) for different resolutions of the initial triangulation $\mathcal{T}^0$. For local barycentric coordinates, the contour line at 10^{-4} is shown in green, and the orange line marks the support of the Loop coordinates. The timings for computing local barycentric coordinates are given at the top.

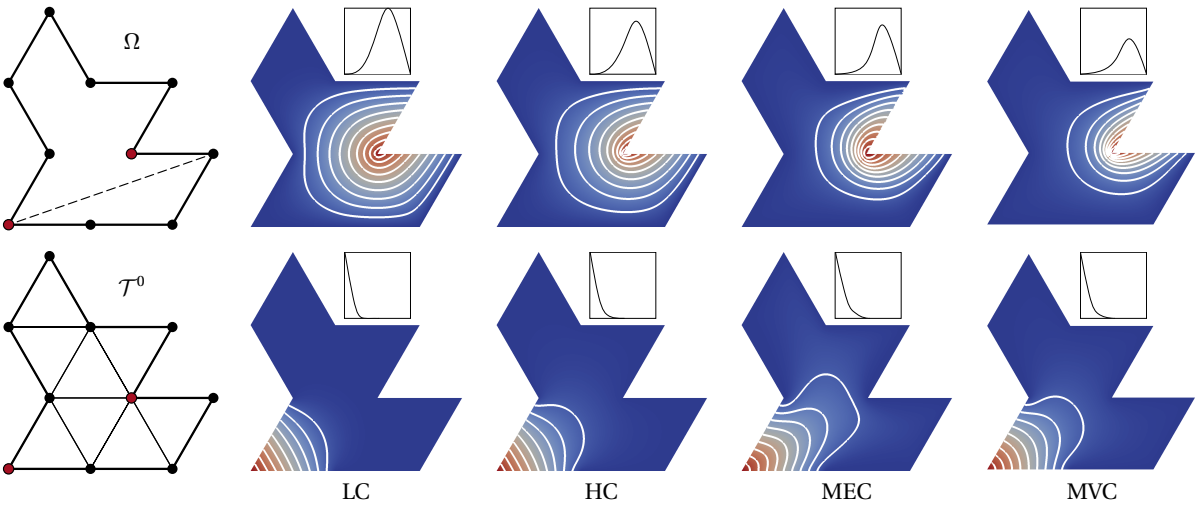


Figure 14: Comparison of different barycentric coordinate functions for the two red vertices. Loop coordinates were computed for the initial triangulation $\mathcal{T}^0$ in the bottom left. The insets show the cross sections along the dashed line.

cost of computing local barycentric coordinates, it is better to smooth them with Loop subdivision instead of further increasing the resolution of the initial triangulation. The fact that the coordinate functions for the triangulation with 27605 vertices look apparently different from the others is probably due to the fact that the solver had not fully converged, even after the indicated 59172 seconds.

For the example in Figure 14, we computed Loop coordinates for a triangulation $\mathcal{T}^0$ without interior vertices, using only the barycentric coordinates in (4) as initial coordinates at the vertices v_i of Ω. Because of the lack of extraordinary interior vertices, the resulting coordinate functions are C^2 in the interior of Ω. The comparison to harmonic (HC), maximum entropy (MEC), and mean value coordinates (MVC) shows that Loop coordinates are more local at convex and less steep at concave corners.

One potential drawback of our approach is that the Loop coordinates b^∞ depend on the initial triangulation $\mathcal{T}^0$. An example of this effect is given in Figure 15, which shows two coordinate functions for two different initial triangulations as well as the difference between them. For convex corners, this difference is usually less than 0.5% and less than 2% for concave corners, but the contour and gradient plots confirm that the global shapes of the coordinate functions are very similar.

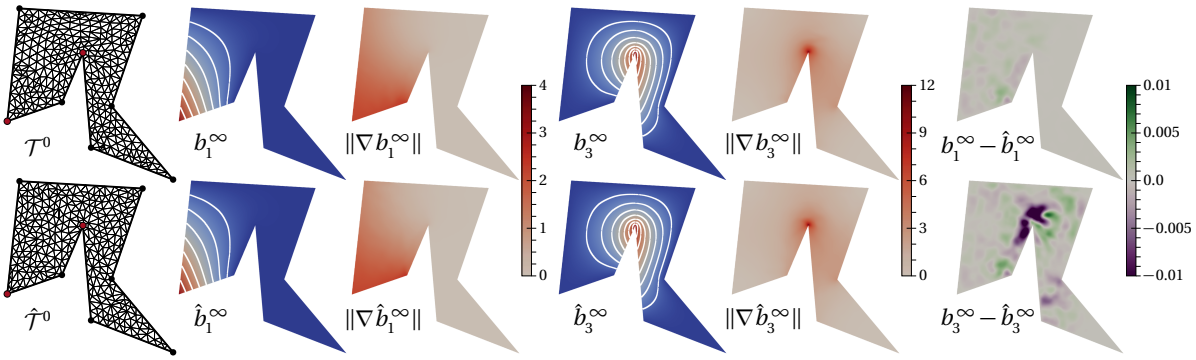


Figure 15: Comparison of barycentric coordinate functions for the two red vertices and different initial triangulations $\mathcal{T}^0$ and $\hat{\mathcal{T}}^0$ using piecewise linear harmonic coordinates as initial coordinates.

4 Catmull–Clark coordinates

The refinement process described in Section 2 works analogously with linear subdivision schemes for quadrilateral meshes. We start from an initial quadrangulation $\mathcal{Q}^0$ of Ω with given barycentric coordinates at the vertices p of $\mathcal{Q}^0$, and use the scheme to generate the sequence of quadrangulations $\mathcal{Q}^0, \mathcal{Q}^1, \ldots$ as well as to compute barycentric coordinates at the vertices of each $\mathcal{Q}^k$. Under the same conditions as in Theorem 4, this gives C^1 continuous barycentric coordinate functions in the limit. The main difference is that we need to be careful with the definition of the functions $b^k = [b_1^k, \ldots, b_n^k]\colon \mathcal{Q}^k \to \mathbb{R}^n$, which interpolate the initial or computed barycentric coordinates at the vertices of $\mathcal{Q}^k$, so as to guarantee the equivalent of Corollary 3. One possibility is to split each quadrilateral of $\mathcal{Q}^k$ into two regular triangles and let b^k be piecewise linear over the triangles obtained this way. Another choice is to let b^k be smooth over each quadrilateral by utilizing mean value coordinates in the following way. For any $p \in \Omega$, let $Q = [p_1, p_2, p_3, p_4]$ be the quadrilateral in $\mathcal{Q}^k$ that contains p, let $\alpha_1, \ldots, \alpha_4$ be the mean value coordinates of p with respect to Q, that is, $p = \sum_{j=1}^4 \alpha_j p_j$, and set $b^k(p) = \sum_{j=1}^4 \alpha_j b^k(p_j)$. Lemma 1 then guarantees that $b^k(p)$ are valid barycentric coordinates of p, and since the weights α_j are non-negative, even for a concave quadrilateral Q [15], the non-negativity statement in Corollary 3 carries over to the quadrilateral setting.

As a case study for this quadrilateral setting we decided to use Catmull–Clark subdivision [3] with the modifications proposed by Biermann et al. [1]. As in Section 3, we mark vertices and edges of Ω as corners and creases to preserve the boundary of the polygon and use the subdivision rules in Figure 16, where the parameters for an interior vertex with valency m are $\alpha_1 = \frac{3}{2m}$ and $\alpha_2 = \frac{1}{4m}$ and the coefficient for the modified edge rules is $\beta = (3 + 2\cos\theta)/8$ with θ as in Section 3. Like Loop coordinates, these *Catmull–Clark* coordinates are C^2 almost everywhere, except at extraordinary interior vertices and convex corners, where they are only C^1 and at concave corners, where they are C^0. To evaluate them, we implemented the same three options as described in Section 3.1 with similar runtimes, using Stam's algorithm [34] for the evaluation of $\boldsymbol{v}$ and $\boldsymbol{b}$ in the interior and the method of Zorin and Kristjansson [43] near the boundary. The vertex

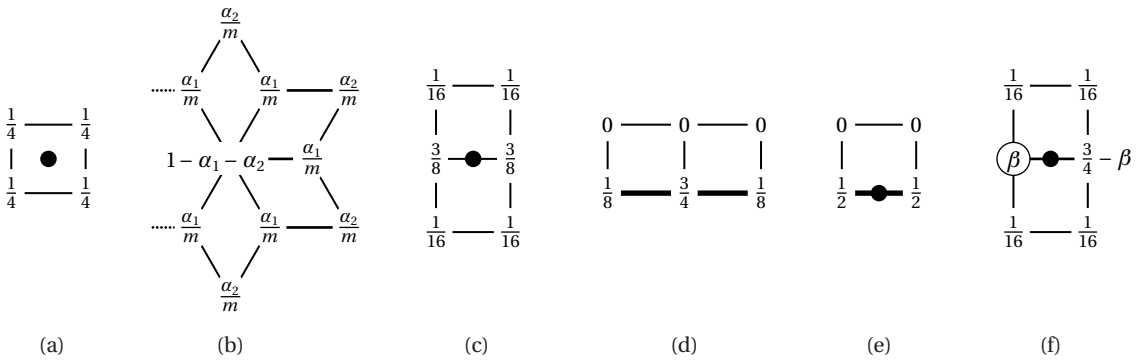


Figure 16: Standard Catmull–Clark subdivision rules for faces (a), interior vertices (b), interior edges (c), boundary vertices (d), boundary edges (e), and modified edge rule (f). Corner vertices are simply interpolated.

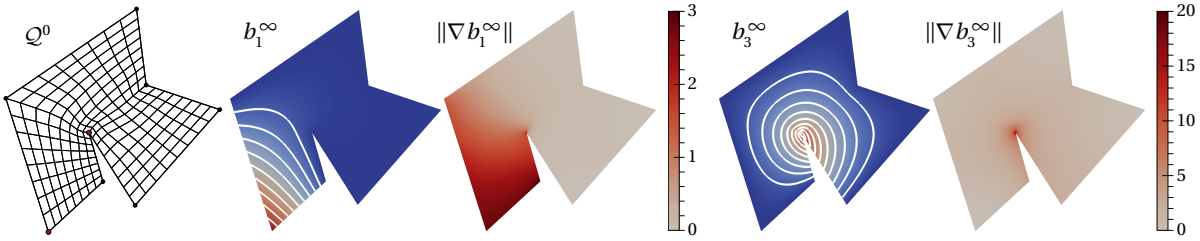


Figure 17: Example of Catmull–Clark coordinate functions and their gradients (shown for the red vertices) with harmonic coordinates over $\mathcal{Q}^0$ as initial coordinates b^0.

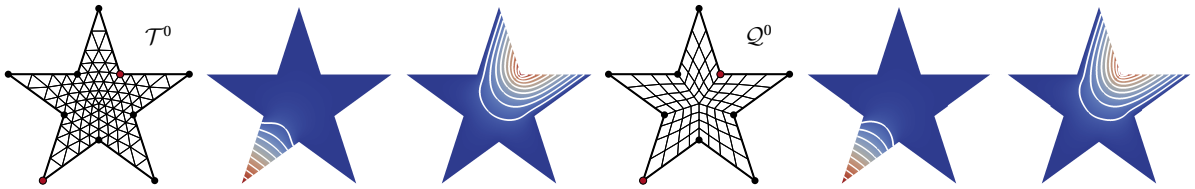


Figure 18: Comparison of Loop (left) and Catmull–Clark (right) coordinate functions (shown for the red vertices) with harmonic coordinates over $\mathcal{T}^0$ and $\mathcal{Q}^0$, respectively, as initial coordinates b^0.

adjustment around a concave corner p is done as in (6) for the adjacent neighbours $p_0,\dots,p_m$ of p, and the opposite corners $q_0,\dots,q_{m-1}$ of the adjacent quadrilaterals are moved to

$$q_i' = p_i' + p_{i+1}' - p, \qquad i = 0,\dots,m-1,$$

so that all adjacent quadrilaterals become congruent parallelograms. With the same arguments as in Theorem 5, one can then show that this configuration scales by a factor of $1/2$ with each subdivision step, thus avoiding foldovers at p, even in the limit. We did not further investigate the issue of internal foldovers, but did not experience any problems in our numerical examples.

4.1 Examples

Figure 17 is the analogue to Figure 10 and shows some examples of Catmull–Clark coordinate functions for harmonic initial coordinates. Since the initial quadrangulation has no extraordinary interior vertices, these functions are C^2 in the interior of Ω, and the plots confirm that they are also visually smooth.

In Figure 18 we computed harmonic coordinates over a triangulation $\mathcal{T}^0$ and a quadrangulation $\mathcal{Q}^0$ of the same polygon and used them as initial coordinates for Loop and Catmull–Clark coordinates, respectively. The example shows that both subdivision schemes have a very similar smoothing effect and that the coordinate functions are almost identical. Since quadrangulating a given polygon is much harder than triangulating it, this suggests that Loop coordinates are probably the method of choice in most cases.

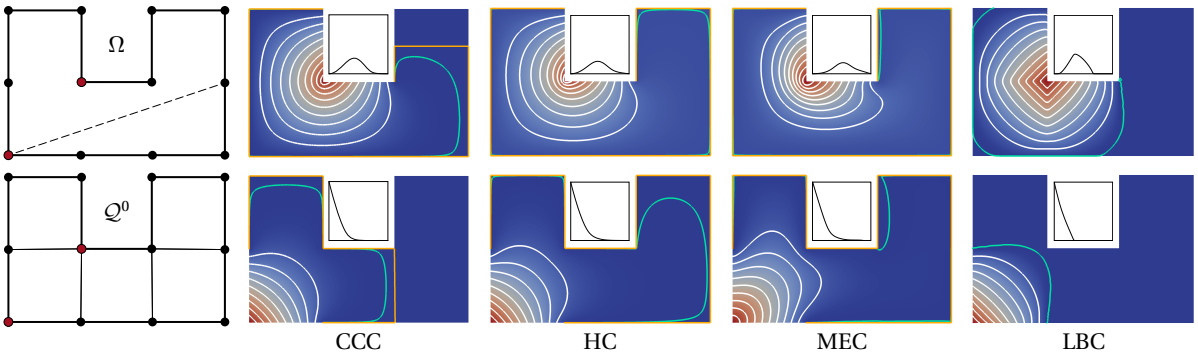


Figure 19: Comparison of different barycentric coordinate functions for the two red vertices. Catmull–Clark coordinates were computed for the initial quadrangulation $\mathcal{Q}^0$ shown in the bottom left. The inset shows the cross section along the dashed line. The contour line at 10^{-4} is shown in green, and the orange line marks the support.

However, for certain polygons like the one in Figure 19 it is more natural to use Catmull–Clark coordinates. Similarly to the example in Figure 14, this figure shows Catmull–Clark coordinates for a quadrangulation $\mathcal{Q}^0$ without interior vertices, using only the barycentric coordinates in (4) as initial coordinates at the vertices of Ω. Consequently, the resulting coordinate functions are C^2 in the interior of Ω. The comparison to harmonic (HC) and maximum entropy coordinates (MEC) shows that Catmull–Clark coordinates (CCC) are more local at convex and less steep at concave corners, and they are smoother, but less local than local barycentric coordinates (LBC).

5 Conclusions

Mesh subdivision is widely known in computer graphics as a technique for creating smooth surfaces with arbitrary topology by repeatedly refining an initial base mesh with simple local rules. In this paper we show that subdivision can also be used to construct barycentric coordinates with favourable properties. While the theory developed in Section 2 is general and works for a large class of subdivision schemes, we believe that Loop subdivision is the method of choice, for two reasons. On the one hand, it is simple and comes with well-understood boundary rules and exact evaluation routines. On the other hand, our examples confirm that the main shape of the limit coordinate functions b_i^∞ is dictated by the initial functions b_i^0, and we do not expect other subdivision schemes to yield qualitatively better results.

However, it still remains future work to develop a strategy for constructing initial triangulations $\mathcal{T}_0$, for which it can be formally proven that the refined triangulations $\mathcal{T}_k$ are regular in the interior, even in the limit. Note that this problem is not restricted to the construction of well-defined Loop coordinates, as it addresses the general question under which conditions the two-dimensional Loop mapping $\boldsymbol{v}: \Omega \to \Omega$ is bijective. Another direction for future work is the extension of our approach to 3D by using volumetric subdivision schemes [4, 30].

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions, which helped to improve this paper. We further thank Teseo Schneider for useful discussions and help with the implementation, and Zishun Liu for providing the local barycentric coordinates data.

References

[1] H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of SIGGRAPH 2000*, Annual Conference Series, pages 113–120, New Orleans, LA, July 2000.

[2] T. J. Cashman. Beyond Catmull–Clark? A survey of advances in subdivision surface methods. *Computer Graphics Forum*, 31(1):42–61, Feb. 2012.

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

[4] Y.-S. Chang, K. T. McDonnell, and H. Qin. A new solid subdivision scheme based on box splines. In K. Lee and N. M. Patrikalakis, editors, *Proceedings of the Seventh ACM Symposium on Solid Modeling an Applications*, pages 226–233, Saarbrücken, Germany, June 2002.

[5] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, Apr. 1990.

[6] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of SIGGRAPH*, pages 173–182, Los Angeles, Aug. 1995.

[7] M. S. Floater. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, Apr. 1997.

[8] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, Mar. 2003.

[9] M. S. Floater and C. Gotsman. How to morph tilings injectively. *Journal of Computational and Applied Mathematics*, 101(1–2):117–129, Jan. 1999.

[10] M. S. Floater, K. Hormann, and G. Kós. A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*, 24(1–4):311–331, Jan. 2006.

[11] I. Ginkel, J. Peters, and G. Umlauf. Normals of subdivision surfaces and their control polyhedron. *Computer Aided Geometric Design*, 24(2):112–116, Feb. 2007.

[12] G. Guennebaud, B. Jacob, et al. Eigen 3.2.7. `http://eigen.tuxfamily.org`, Nov. 2015. [Online; accessed 13-January-2016].

[13] K. Hormann and M. S. Floater. Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics*, 25(4):1424–1441, Oct. 2006.

[14] K. Hormann and N. Sukumar. Maximum entropy coordinates for arbitrary polytopes. *Computer Graphics Forum*, 27(5):1513–1520, July 2008. Proceedings of SGP 2008.

[15] K. Hormann and M. Tarini. A quadrilateral rendering primitive. In T. Akenine-Möller and M. McCool, editors, *Graphics Hardware 2004*, Eurographics Symposium Proceedings, pages 7–14, Grenoble, France, Aug. 2004.

[16] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Transactions on Graphics*, 26(3):Article 71, 9 pages, July 2007. Proceedings of SIGGRAPH 2007.

[17] T. Ju, S. Schaefer, and J. Warren. Mean value coordinates for closed triangular meshes. *ACM Transactions on Graphics*, 24(3):561–566, July 2005. Proceedings of SIGGRAPH 2005.

[18] X.-Y. Li and S.-M. Hu. Poisson coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):344–352, Feb. 2013.

[19] X.-Y. Li, T. Ju, and S.-M. Hu. Cubic mean value coordinates. *ACM Transactions on Graphics*, 32(4):Article 126, 10 pages, July 2013. Proceedings of SIGGRAPH 2013.

[20] Y. Lipman, J. Kopf, D. Cohen-Or, and D. Levin. GPU-assisted positive mean value coordinates for mesh deformations. In A. Belyaev and M. Garland, editors, *Proceedings of SGP 2007*, Eurographics Symposium Proceedings, pages 117–123, Barcelona, Spain, July 2007.

[21] C. T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, The University of Utah, Aug. 1987.

[22] C. T. Loop and T. D. DeRose. A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics*, 8(3):204–234, July 1989.

[23] J. Manson, K. Li, and S. Schaefer. Positive Gordon–Wixom coordinates. *Computer-Aided Design*, 43(11):1422–1426, Nov. 2011. Proceedings of SPM 2011.

[24] J. Manson and S. Schaefer. Moving least squares coordinates. *Computer Graphics Forum*, 29(5):1517–1524, July 2010. Proceedings of SGP 2010.

[25] S. Martin, P. Kaufmann, M. Botsch, M. Wicke, and M. Gross. Polyhedral finite elements using harmonic basis functions. *Computer Graphics Forum*, 27(5):1521–1529, July 2008.

[26] M. Meyer, H. Lee, A. Barr, and M. Desbrun. Generalized barycentric coordinates on irregular polygons. *Journal of Graphics Tools*, 7(1):13–22, 2002.

[27] A. F. Möbius. *Der barycentrische Calcul.* Johann Ambrosius Barth, Leipzig, 1827.

[28] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Series in Operations Research. Springer, New York, 1999.

[29] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.

[30] S. Schaefer, J. Hakenberg, and J. Warren. Smooth subdivision of tetrahedral meshes. In R. Scopigno and D. Zorin, editors, *Proceedings of SGP 2004*, Eurographics Symposium Proceedings, pages 147–154, Nice, France, July 2004.

[31] S. Schaefer and J. Warren. Exact evaluation of non-polynomial subdivision schemes at rational parameter values. In M. Alexa, S. Gortler, and T. Ju, editors, *Proceedings of Pacific Graphics 2007*, pages 321–330, Maui, HI, Oct./Nov. 2007.

[32] S. Schaefer and J. Warren. Exact evaluation of limits and tangents for non-polynomial subdivision schemes. *Computer Aided Geometric Design*, 25(8):607–620, Nov. 2008.

[33] J. R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In M. C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, Berlin, Heidelberg, 1996.

[34] J. Stam. Exact evaluation of Catmull–Clark subdivision surfaces at arbitrary parameter values. In *Proceedings of SIGGRAPH 98*, Annual Conference Series, pages 395–404, Orlando, FL, July 1998.

[35] J. Stam. Exact evaluation of Loop subdivision surfaces. In *Proceedings of SIGGRAPH 98*, CD-ROM, pages 1–15, Orlando, FL, July 1998.

[36] G. Strang and G. Fix. *An Analysis of The Finite Element Method.* Wellesley-Cambridge Press, 2nd edition, 2008.

[37] N. Sukumar and E. A. Malsch. Recent advances in the construction of polygonal finite element interpolants. *Archives of Computational Methods in Engineering*, 13(1):129–163, Mar. 2006.

[38] E. L. Wachspress. *A Rational Finite Element Basis*, volume 114 of *Mathematics in Science and Engineering*. Academic Press, New York, 1975.

[39] S. Waldron. Affine generalised barycentric coordinates. *Jaen Journal on Approximation*, 3(2):209–226, 2011.

[40] J. Warren, S. Schaefer, A. Hirani, and M. Desbrun. Barycentric coordinates for convex sets. *Advances in Computational Mathematics*, 27(3):319–338, Oct. 2007.

[41] O. Weber and C. Gotsman. Controllable conformal maps for shape deformation and interpolation. *ACM Transactions on Graphics*, 29(4):Article 78, 11 pages, July 2010. Proceedings of SIGGRAPH.

[42] J. Zhang, B. Deng, Z. Liu, G. Patanè, S. Bouaziz, K. Hormann, and L. Liu. Local barycentric coordinates. *ACM Transactions on Graphics*, 33(6):Article 188, 12 pages, Nov. 2014. Proceedings of SIGGRAPH Asia 2014.

[43] D. Zorin and D. Kristjansson. Evaluation of piecewise smooth subdivision surfaces. *The Visual Computer*, 18(5–6):299–315, Aug. 2002.

[44] D. Zorin and P. Schröder. *Subdivision for Modeling and Animation*. Number 23 in SIGGRAPH 2000 Course Notes. ACM Press, July 2000.

[45] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH 1996*, Annual Conference Series, pages 189–192, New Orleans, LA, Aug. 1996.