Tessellation of Zheng–Ball patches

Péter Salvi and Patrik Gál

Budapest University of Technology and Economics

Abstract

Some of the earliest multi-sided surface formulations depend on a parameterization defined implicitly by constraints. This presents a difficulty in visualizing the patches, as the parametric domain is not trivial to triangulate. We propose a general method that can be used with any number of sides.

1. Introduction

In recent years, n-sided surface representations have received renewed attention.[9, 1] These formulations can describe non-four-sided free-form surfaces without the inaccuracies of the usual *trimming* approach. An early example of such patches was proposed by Sabin,[5] where $2n+1$ control points were used to define three- and five-sided surfaces that could connect to quadratic four-sided patches with G^1 continuity. The same idea was extended later to six-sided surfaces and cubic boundaries,[3] and then to a general degree.[10]

These surfacing methods depend on n scalar parameters defined implicitly by constraints, representing a kind of distance from the boundaries. Consequently, the patch equation is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^3$, and to visualize the surface, we first need a tessellation of the n-dimensional parameter values. It is not evident how to do this, and this paper explores different options.

2. Preliminaries

Parameters $\mathbf{u} = \{u_j\}_1^n \in [0,1]^n$ form a 2-dimensional surface in $\mathbb{R}^n$ space. The side of the domain where $u_i = 0$ is mapped to the i^{th} boundary of the patch. At these points, the following constraints apply:

$$u_{i-1} + u_{i+1} = 1, \tag{1}$$

$$u_j = 1, \qquad j \neq i-1, i, i+1 \tag{2}$$

$$\sum_j u'_j = c_n u_{i-1} u_{i+1}, \tag{3}$$

$$u_{i+1} u'_{i-1} = u_{i-1} u'_{i+1}, \qquad \text{when } n > 3 \tag{4}$$

where c_n is a constant depending only on the number of sides, and derivatives are with respect to u_i. The patch equation using these parameters can be found in the Appendix.

3. Tessellation

In the following we will review the parameterizations of three-, five- and six-sided patches, and propose tessellation algorithms of increasing complexity to handle the arising difficulties.

3.1. Three-sided patches

The three-sided case is the simplest. Using the normalization equation[5]

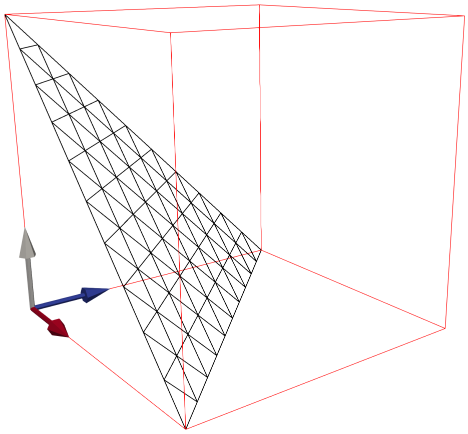
$$u_1 + u_2 + u_3 - 2u_1u_2u_3 = 1, \tag{5}$$

we can choose $c_3 = 2$, $u'_{i-1} = -u_{i-1}^2$ and $u'_{i+1} = -u_{i+1}^2$ to satisfy Eqs. (1)–(3). By Eq. (2) we also know that corners of the domain have parameters $(1,0,0)$, $(0,1,0)$, $(0,0,1)$ – generally sequences of 1s with two consecutive 0s at the indices associated with the two boundaries at the corner. We can create a linear triangulation connecting these three vertices, see Fig. 1a.

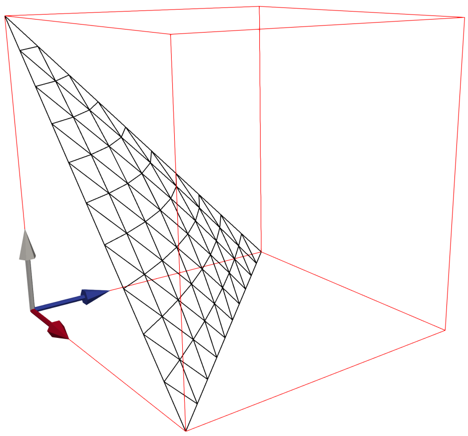
Now we can take two coordinates, e.g. u_1 and u_2, and express the third by them using Eq. (5):

$$\hat{u}_3 = \frac{1 - u_1 - u_2}{1 - 2u_1u_2}. \tag{6}$$

The $(u_1, u_2, \hat{u}_3)$ points are on the domain surface, see Figure 1b. The difference from the linear (barycentric) parameters can be seen clearly on Fig. 2, where the domain is depicted from the side. Note that the triangulation will not be symmetric, since we arbitrarily chose $\hat{u}_3$ to express from the other two, but this is not important at sufficiently high resolutions.



(a) Linear tessellation



(b) Constrained parameters

Figure 1: Tessellation of a triangular domain.

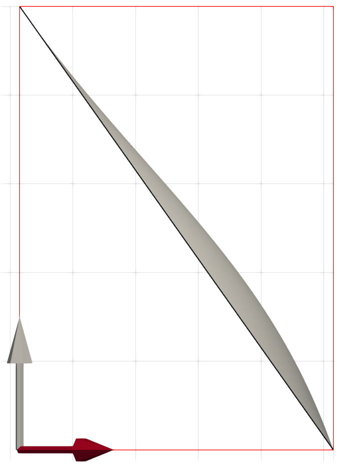


Figure 2: Difference between Figs. 1a and 1b.

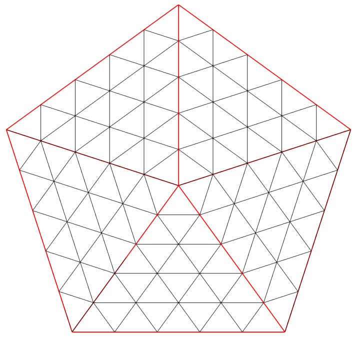


Figure 3: Tessellation of a 5-sided polygon.

3.2. Five- and six-sided patches

For five-sided patches, we have the normalization equations[5]

$$u_j = 1 - u_{j+2}u_{j+3}, \qquad j = 1\dots 5 \tag{7}$$

using cyclic indexing. At first sight, this defines 5 equations for the 5 parameters, but actually only 3 of the equations are independent, so we still get a two-dimensional surface in $\mathbb{R}^5$. Setting $c_5 = 1$, $u'_{i-1} = u_{i-1}^2 u_{i+1}$ and $u'_{i+1} = u_{i+1}^2 u_{i-1}$, it is easy to see that Eqs. (1)–(4) hold.

An n-sided polygon can be tessellated by first dividing it into n triangles, see Fig. 3. For this, we need a center position. It is natural to require that at the middle of the surface all parameters should be equal. Simple algebra shows that $(\varphi,\varphi,\varphi,\varphi,\varphi)$ is actually on the domain surface, where $\varphi = (\sqrt{5}-1)/2$ is the golden ratio.

We can create a linear triangulation of each sub-triangle, as in the previous section. Choosing u_1 and u_2 as base, the rest can be computed as

$$\hat{u}_3 = \frac{1-u_1}{1-u_1u_2}, \quad \hat{u}_4 = 1 - u_1u_2, \quad \hat{u}_5 = \frac{1-u_2}{1-u_1u_2}, \tag{8}$$

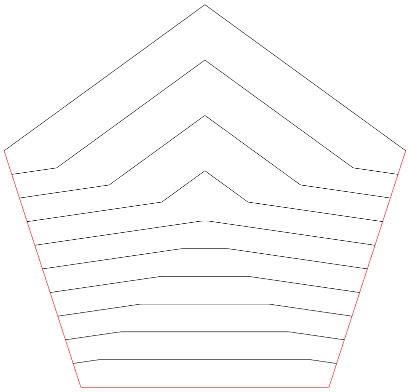
but then the sub-triangle associated with the $u_4 = 0$ boundary will be degenerate, as on that side $u_1 = u_2 = 1$.

It is better to "project" the linear triangulation onto the domain surface by minimizing the *normalization energy*

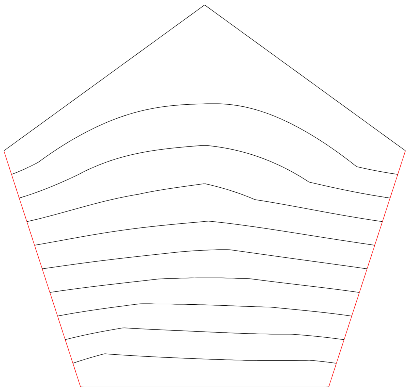
$$E_5(\mathbf{u}) = \sum_{j=1}^{5} \left(u_j + u_{j+2}u_{j+3} - 1\right)^2, \tag{9}$$

which is just the sum of squared normalization errors based on Eq. (7). This can be minimized using a simple derivative-free optimization algorithm, such as Powell's method.[4] Figure 4 shows constant parameter lines before and after the projection. While these show various breaks, remember that this represents only one coordinate of a continuous 5-dimensional surface.

Instead of starting from a linear approximation, we can also make use of generalized barycentric coordinates.[2] First we tessellate a regular n-sided polygon (as in Fig. 3), then

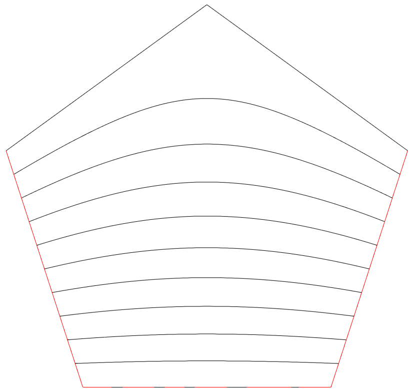


(a) Linear approximation

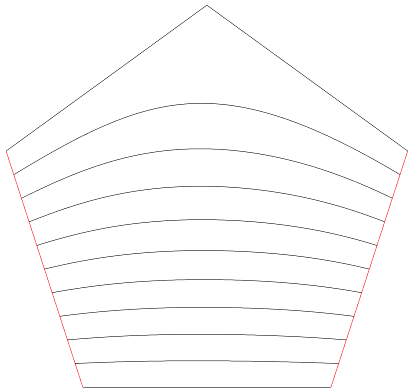


(b) After projection

Figure 4: Parameter lines associated with the bottom side.



(a) Wachspress-based approximation



(b) After projection

Figure 5: Projection by generalized barycentric coordinates.

map the vertices to (e.g.) Wachspress coordinates $\mathbf{w} = \{w_j\}_1^n$. We can use the *distance parameter*[8] $u_i = 1 - w_{i-1} - w_i$ as our first approximation, since this also satisfies Eqs. (1)–(2). Finally the vertices are projected onto the domain as above, so that Eqs. (3)–(4) will also hold. Figure 5 shows constant parameter lines for this method. Note that the parameters in Figures 4 and 5 define *the same* surface.

Generalized baryentric coordinates can also be used with the direct method of Eq. (8). The $u_4 = 0$ side will still be degenerate (we can use $\hat{u}_4 = u_4$ there) but other parts of the sub-triangle behave normally.

In the case of six-sided patches, there is no rational formula for the parameters.[6] The normalization equations (only four of which are independent) are:[7]

$$u_{i+1}^2(1 - u_{i-1}u_i)(1 - 2u_{i-1}u_i) + u_{i+1}(2u_{i-1} - 3u_{i-1}^2 u_i + u_{i-1}u_i^2) + u_{i-1}^2 = 1. \tag{10}$$

Setting $c_6 = 1$ with

$$u'_{i-2} = \frac{1}{2}u_{i-1}u_{i+1} - u_{i+1}, \tag{11}$$

$$u'_{i-1} = \frac{3}{2}u_{i-1}^2 u_{i+1}, \tag{12}$$

$$u'_{i+1} = \frac{3}{2}u_{i-1}u_{i+1}^2, \tag{13}$$

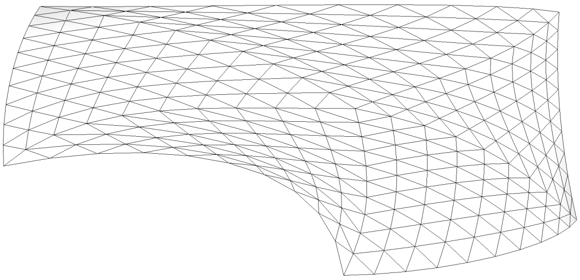
$$u'_{i+2} = \frac{1}{2}u_{i-1}u_{i+1} - u_{i-1}, \tag{14}$$

$$u'_{i+3} = -\frac{1}{2}u_{i-1}u_{i+1}, \tag{15}$$

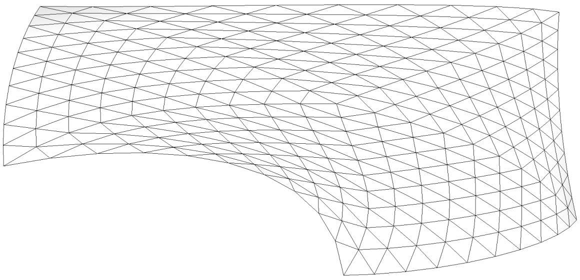
all the requirements are satisfied. Since the direct method would involve solving a non-linear equation, it is simpler to use projection here, as well. The center point in this case is $(\psi, \psi, \psi, \psi, \psi, \psi)$ with $\psi = \frac{1}{\sqrt{2}}$.

4. Results

Figure 6 shows a 5-sided patch triangulated with the projection method, starting from both the linear and Wachspress-based approximations, with 10 triangles on each boundary. Generalized Bézier patches[9] share the same control struc-



(a) Projection from linear approximation



(b) Projection from Wachspress approximation

Figure 6: Triangulated surfaces.

ture, see Fig. 7. A comparison is shown on Fig. 8, with a linear-based dense triangulation (50 triangles per boundary). The patches are generated from the same control points. The isophote lines are very similar, and match at the boundaries (as required).

Conclusion

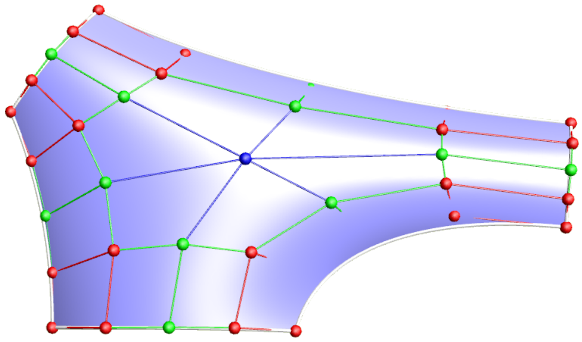
We have shown different methods for the tessellation of Zheng–Ball patches. Of these, the only one that scales to arbitrary number of sides is the projection method. As future work, gradient descent could be used instead of Powell's method, which would result in increased symmetry. Parameterizations for more sides should also be investigated, probably using biharmonic surfaces.

Acknowledgements

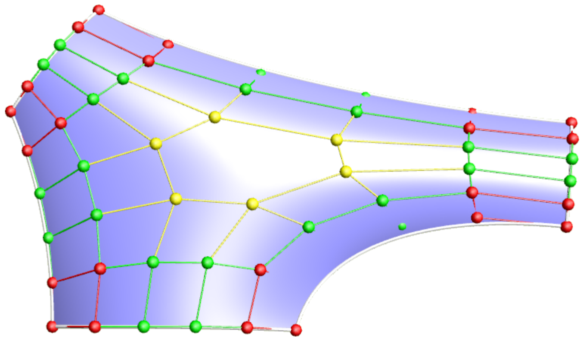
The authors thank Márton Vaitkus for the fruitful discussions on this subject. This project has been supported by the Hungarian Scientific Research Fund (OTKA, No.124727).

Appendix A: The Zheng–Ball patch equation

All control points are given indices based on their position in the control structure. An index $\boldsymbol{\lambda} = \{\lambda_j\}_1^n$ is composed of n natural numbers, with λ_j signifying how far removed is the control point from the i^{th} side. For example, a control point on side i has $\lambda_i = 0$; the central control point in Fig. 7a has the index $(2,2,2,2,2,2)$. Generally, in a patch of $n > 3$



(a) Even degrees



(b) Odd degrees

Figure 7: Zheng–Ball / Generalized Bézier control net.

sides with degree-d boundaries, when the $(i-1)^{\text{st}}$ and i^{th} sides are closest to a control point, the remaining indices are computed as

$$\lambda_{i-2} = d - \lambda_i, \quad \lambda_{i+1} = d - \lambda_{i-1}, \quad \lambda_j = d - \min(\lambda_{i-1}, \lambda_i),$$

for $j \notin \{i-2, i-1, i, i+1\}$.

The patch is evaluated at parameter $\mathbf{u}$ as

$$S(\mathbf{u}) = \sum_{\boldsymbol{\lambda}} P_{\boldsymbol{\lambda}} B_{\boldsymbol{\lambda}}(\mathbf{u}),$$

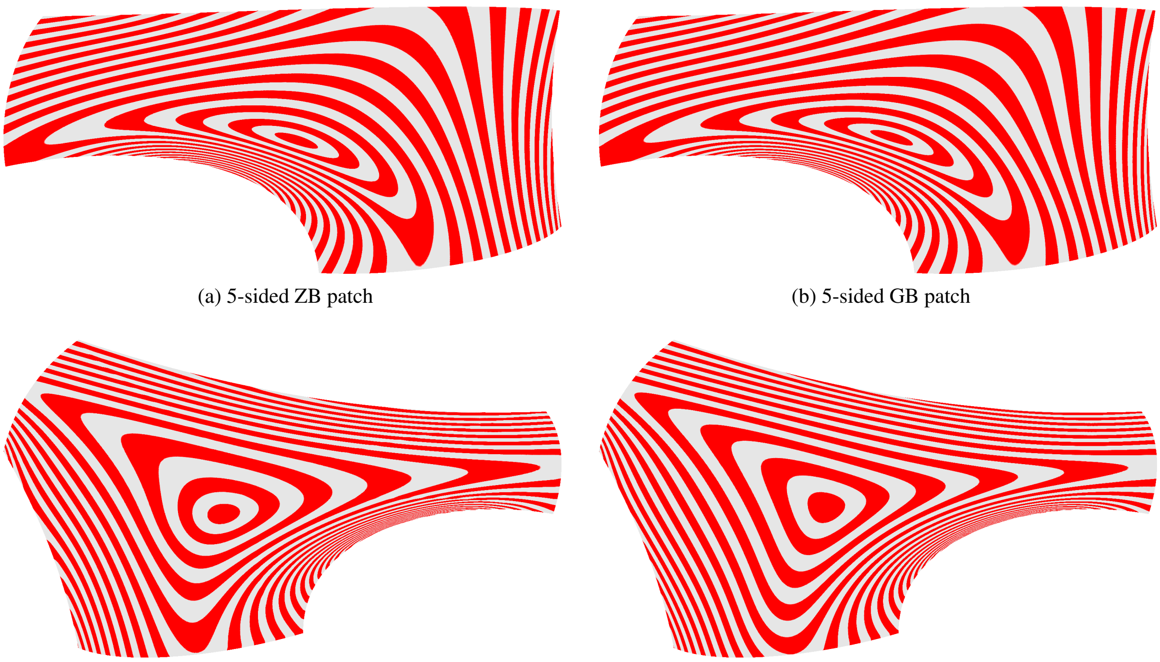
where $P_{\boldsymbol{\lambda}}$ denotes the control points, and $B_{\boldsymbol{\lambda}}(\mathbf{u})$ is a suitable blending function.

For control points *not* on the boundary (i.e., $\forall j : \lambda_j > 0$) we define the following blend:

$$B^*_{\boldsymbol{\lambda}}(\mathbf{u}) = \binom{d}{\lambda_{i-1}} \binom{d}{\lambda_i} \prod_{j=1}^{n} u_j^{\lambda_j},$$

where λ_{i-1} and λ_i are the smallest elements in $\boldsymbol{\lambda}$. For control points on the i^{th} boundary, we have $\lambda_{i-1} = k$, $\lambda_i = 0$, $\lambda_{i+1} = d-k$, and all other values in $\boldsymbol{\lambda}$ are 1. In this case, the blend is defined as

$$B^*_{\boldsymbol{\lambda}}(\mathbf{u}) = \binom{d}{k} u_{i-1}^k u_{i+1}^{d-k} \prod_{\substack{j=1 \\ j \neq i-1,i,i+1}}^{n} u_j^d \left(1 - dc_n \prod_{j=1}^{n} u_j\right).$$



(a) 5-sided ZB patch

(b) 5-sided GB patch

(c) 6-sided ZB patch

(d) 6-sided GB patch

Figure 8: Comparison of Zheng–Ball (ZB) and Generalized Bézier (GB) patches with the same control points.

This latter equation is a bit different for triangular patches:

$$B^*_{\boldsymbol{\lambda}}(\mathbf{u}) = \binom{d}{k} u_{i-1}^k u_{i+1}^{d-k} \left(1 - u_i f_{\boldsymbol{\lambda}}(\mathbf{u})\right),$$

where

$$f_{\boldsymbol{\lambda}}(\mathbf{u}) = \begin{cases} d-k+(2k-d)u_{i+1} & \text{when } d-k \le k, \\ k+(d-2k)u_{i-1} & \text{when } d-k > k. \end{cases}$$

Unfortunately $B^*_\Sigma(\mathbf{u}) = \sum_{\boldsymbol{\lambda}} B^*_{\boldsymbol{\lambda}}(\mathbf{u})$ does not equal to 1, which would be needed for the surface to be an affine combination of the control points, so we distribute the weight deficiency $(1 - B^*_\Sigma(\mathbf{u}))$ between the blending functions associated with control points *not* on the boundary:

$$B_{\boldsymbol{\lambda}}(\mathbf{u}) = \begin{cases} B^*_{\boldsymbol{\lambda}}(\mathbf{u}) & \text{when } \exists j : \lambda_j = 0, \\ B^*_{\boldsymbol{\lambda}}(\mathbf{u}) + \frac{1 - B^*_\Sigma(\mathbf{u})}{T_{n,d}} & \text{otherwise,} \end{cases}$$

where

$$T_{n,d} = \begin{cases} \frac{n(d-2)d}{4} + 1 & \text{when } d \text{ is even,} \\ \frac{n(d-1)^2}{4} & \text{when } d \text{ is odd} \end{cases}$$

is the number of control points *not* on the boundary.

References

1. Gerben J. Hettinga and Jiří Kosinka. Multisided generalisations of Gregory patches. *Computer Aided Geometric Design*, 62:166–180, 2018.
2. Kai Hormann and N. Sukumar. *Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics*. CRC press, 2017.
3. Mamoru Hosaka and Fumihiko Kimura. Non-four-sided patch expressions with control points. *Computer Aided Geometric Design*, 1(1):75–86, 1984.
4. Mykel J. Kochenderfer and Tim A. Wheeler. *Algorithms for Optimization*. MIT Press, 2019.
5. Malcolm A. Sabin. Non-rectangular surface patches suitable for inclusion in a B-spline surface. In *Proceedings of the Eurographics Conference*, pages 57–69, 1983.
6. Malcolm A. Sabin. Some negative results in n sided patches. *Computer-Aided Design*, 18(1):38–44, 1986.
7. Malcolm A. Sabin. A symmetric domain for a 6-sided patch. In *The Mathematics of Surfaces IV*, pages 185–193, 1994.
8. Tamás Várady, Péter Salvi, and György Karikó. A multi-sided Bézier patch with a simple control structure. *Computer Graphics Forum*, 35(2):307–317, 2016.
9. Tamás Várady, Péter Salvi, and István Kovács. Enhancement of a multi-sided Bézier surface representation. *Computer Aided Geometric Design*, 55:69–83, 2017.
10. Jinjin Zheng and Alan A. Ball. Control point surfaces over non-four-sided areas. *Computer Aided Geometric Design*, 14(9):807–821, 1997.