Seventh Hungarian Conference on Computer Graphics and Geometry, Budapest, 2014

Multi-sided Surfaces with Curvature Continuity

Péter Salvi, Tamás Várady

Budapest University of Technology and Economics

Abstract

The basic idea of curve network-based design is to construct a collection of smoothly connected surface patches that interpolate boundary constraints extracted solely from the curve network. While the majority of applications demands only tangent plane (G^1) continuity between the adjacent patches, there are applications where curvature continuous connections (G^2) are required. Examples include handling special curve network configurations with supplemented internal edges, "master-slave" curvature constraints and general topology surface approximations over meshes. The first step of the surface generation process is the construction of interpolant surfaces that enforce suitable cross-derivatives for transfinite surface patches; these interpolants are often called ribbons. *For G^2 interpolation we extend Gregory's multi-sided surface scheme, and focus on creating and combining special* parabolic *ribbons. We discuss the basic patch construction including the blending functions and a special sweepline parameterization. A proof of G^2 continuity is given in the Appendix. The application of curvature continuous multi-sided patches is demonstrated by a few simple examples.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling — curvenet-based design, transfinite surfaces, Gregory patches, G^2 continuity

1. Introduction

In curve network-based design, surface models are directly defined by a collection of freeform curves, arranged into a single 3D network with general topology. Curves may come from (i) sketch input, (ii) feature curves extracted from orthogonal views, (iii) curves traced on triangular meshes or (iv) direct 3D editing. Once the curves are defined, all the surfaces are generated automatically. This calls for a representation based on geometric information extracted solely from the boundaries. Transfinite surface interpolation is a natural choice, as it does not require a grid of control points to define the interior shape, and all n boundaries are handled uniformly, unlike in the case of trimmed quadrilateral surfaces. The ability to interactively edit prescribed boundaries and cross-derivatives is also an advantage in contrast to recursive subdivision schemes.

The first step of surface generation is to compute cross-directional data, such as common tangent planes and, when needed, curvatures that will be shared by the adjacent patches. Then interpolant surfaces or *ribbon*s are generated, that carry first or second-degree cross-derivative constraints to be eventually interpolated by the transfinite surfaces.

The majority of multi-sided transfinite surfaces are defined over convex domains, combining only linear ribbon surfaces and enabling G^1 continuity between the adjacent patches. At the same time, there are several practical design situations, where this approach is not sufficient, and higher degree continuity is required.

(i) It often occurs that additional curves need to be inserted into the curve network to make it suitable for applying convex methods. The supplemented curves must be compatible with the already defined ribbons, and it is particularly important to produce seamless transitions along these curves. A typical example is when two curves span a concave angle at a common vertex and then a composite patch — with convex domains — is created. Another example is to generate surfaces by interpolating two disjoint loops with prescribed slopes (see Figure 7 later in Section 5).

(ii) Another interesting situation is when a designer wants to retain one of the surfaces (the *master*), but also wants to prescribe a curvature continuous connection for the adjacent patch (the *slave*), see Figure 8.

(iii) A third example is when we have a general topology curve network defined over a mesh and want to obtain a good approximation of the interior data points (see Figure 9). Clearly, if we extract not only the normal vectors, but also curvature information from the underlying mesh, we can obtain a more accurate approximation.

In the above cases, it is possible to ensure G^2 continuity by using parabolic ribbons. In Section 2 we give an overview of related publications. In Section 3 various methods to compute linear and parabolic interpolants will be discussed, while in Section 4 we introduce an extended Gregory patch formulation with a new sweepline parameterization that makes it possible to combine parabolic ribbons and yield curvature continuous surfaces. A few examples will illustrate the surface scheme in Section 5.

2. Previous Work

Most papers concerning transfinite surface interpolation assume that either the ribbons, or at least the cross-derivatives are given, and very little is written about how this information can be extracted from a curve network. The reason for this apparent lack of interest may be that at the time when transfinite surface interpolation appeared with the Coons[2] and Gregory[1] patches, its main application was hole filling and vertex blending, where additional cross-directional data were readily available. This trend lived on, even after the advent of curve network-based ideas, such as the Minimum Variation Surfaces[9].

There is a constant interest in multi-sided patches with G^2 continuity, the first efforts dating back 25 years[3]. In particular, an extension of the Gregory patch similar to the one described here was published by Hall and Mullineux[4].

Special treatment of difficult configurations is also a recurring theme in the literature; most approaches use concave domains[7, 8, 6] to handle holes and extreme spatial structures, though we have very limited information concerning the practical design potential and the quality of these shapes.

3. Computing Ribbons

In the course of curve network-based design, users create and/or edit the boundary curves, while surfaces are generated in an automatic manner. This means that cross-derivatives and curvatures are also derived solely from the curve network. Here we will deal only with generating smooth connections, though in a real system the user may want to create sharp edges, as well.

All continuity constraints are accomplished through defining “proper” ribbons, i.e., once we set the adjacent ribbons G^1 or G^2 continuous, the respective transfinite surfaces will inherit this property.

Two adjacent ribbons are G^1 continuous if they share the same sweep of normal vectors (called the *normal fence*) along the common boundary. Figure 1 shows an example, where the fence is rendered as a series of yellow lines. The necessary condition for G^1 continuity is that the first cross-derivatives on both sides are perpendicular to the fence.

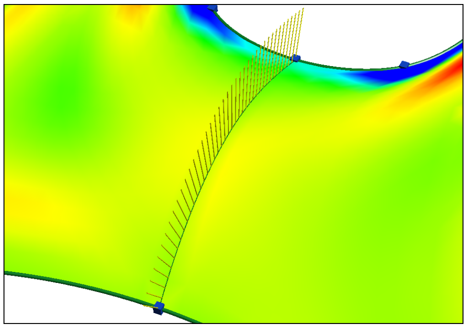


Figure 1: Normal fence and mean curvature map

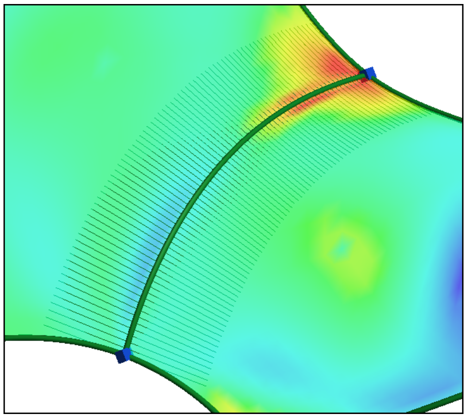


Figure 2: Normal curvature arcs and Gauss curvature map

When we require G^2 continuity between two surfaces, we can make use of the Linkage Curve Theorem[10, 5]:

> Two surfaces tangent along a C^1-smooth linkage curve are curvature continuous, if and only if at every point of the linkage curve, their normal curvature agrees for an arbitrary direction other than the tangent of the linkage curve.

This practically means that if we have a particular directional sweep along the common boundary, and the normal curvatures of the two ribbon surfaces in this direction are always the same, then we have G^2 continuity. Figure 2 shows an example with the common normal curvatures shown as circular arcs.

In the rest of this section, we will investigate how to determine normals and curvatures from the curve network.

3.1. Preliminaries

A ribbon is a four-sided side interpolant surface $R(s,d)$, where s is the *side parameter*, and d is the *distance parameter*. The side parameter is in the interval $[0,1]$ and runs along the boundary curve. The distance parameter is defined in the cross direction; it is zero on the boundary and increases as we move away from it.

For a given n-sided patch, there is a loop of curves, $P_i(s_i)$, and we need to create the corresponding ribbons $R_i(s_i,d_i)$, $i \in [1 \dots n]$. The parameters (s_i,d_i) can also be regarded as functions that map values from a common domain (see Section 4.2). Note also that the indexing is circular, with 1 coming after n and vice versa, and we have $P_{i-1}(1) = P_i(0)$ for all i.

3.2. Cross-derivatives

We assume that for each vertex of the network the crossing curves define a local tangent plane. For each boundary $P_i(s_i)$ of a given patch, there exists a normal fence $N_i(s_i)$ that interpolates the normals at the related corners and minimizes its rotation along the boundary. This is called a rotation-minimizing frame or RMF. An exact (closed form) solution of the underlying differential equation cannot be determined, but approximations can be computed via a sequence of discrete points[14].

The cross-derivatives of the ribbons are defined as

$$\begin{aligned} T_i(s_i) &:= \frac{\partial}{\partial d_i} R_i(s_i,0) \\ &= \alpha(s_i) D_i(s_i) + \beta(s_i) \frac{\partial}{\partial s_i} R_i(s_i,0), \end{aligned}$$

where $\alpha(s_i)$ and $\beta(s_i)$ are scalar functions, and $D_i(s_i)$ represents a direction vector function, that is perpendicular to $N_i(s_i)$ everywhere. One trivial choice is the binormal

$$D_i(s_i) = N_i(s_i) \times \frac{\partial}{\partial s_i} R_i(s_i,0),$$

but other definitions are also possible.

The scalar functions satisfy end conditions at the corner points ($s_i = 0$ and $s_i = 1$), but there are further degrees of freedom to define these in order to optimize the shape of the patch. For example, cross-derivatives at the middle of the boundary ($s_i = 0.5$) can be prescribed by users for enhanced control of the surface shape. Alternatively, these can be optimized by fairing algorithms, which is the subject of ongoing research.

3.3. Creating Ribbons

We will look at two types of side interpolants: linear ribbons, that are ruled surfaces suitable for creating a G^1-continuous model, and parabolic ribbons, that are quadratic in the cross direction, and rely on the computed common curvature values to ensure G^2 continuity.

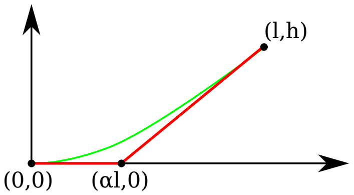


Figure 3: Setting the control points of a parabolic ribbon.

3.3.1. Linear Ribbons

Given a boundary curve $P_i(s_i)$ and the corresponding cross-derivative $T_i(s_i)$, ribbon construction is straightforward:

$$R_i(s_i,d_i) = P_i(s_i) + d_i T_i(s_i).$$

3.3.2. Parabolic Ribbons

Parabolic ribbons are also simple, having the form

$$R_i(s_i,d_i) = P_i(s_i) + d_i T_i(s_i) + \frac{1}{2} d_i C_i(s_i),$$

where $C_i(s_i)$ is the second cross-derivative of the ribbon.

It is a natural choice to calculate the normal curvature in the sweeping direction of the first cross-derivative, i.e., in the plane spanned by $T_i(s_i)$ and $N_i(s_i)$. Let us transform, for ease of computation, the parabolic arc of R_i at a fixed $\hat{s}_i$ into a local coordinate system, where the first point is the origin, and the tangent of the arc is the local x-axis. Then at a given boundary point the equation of the parabola can be written as a quadratic Bézier curve

$$R_i(\hat{s}_i,d_i) = B_0^2(d_i) \cdot (0,0) + B_1^2(d_i) \cdot (\alpha l,0) + B_2^2(d_i) \cdot (l,h),$$

yielding $\kappa = \frac{h}{2\alpha^2 l^2}$ (see Figure 3). Assuming that the width of the parabolic ribbon is the same as the corresponding linear one, l is already defined. Then the prescribed curvature κ can be set by means of α and h, which define the second and third control points of the parabolic arc.

The choice of α gives us a degree of freedom. It may be chosen as a constant. A better choice is to optimize α so that the parabolic ribbon should minimally deviate from the corresponding linear one. This can be formalized as

$$\left(\alpha l - \frac{l}{2}\right)^2 + h^2 \to \min,$$

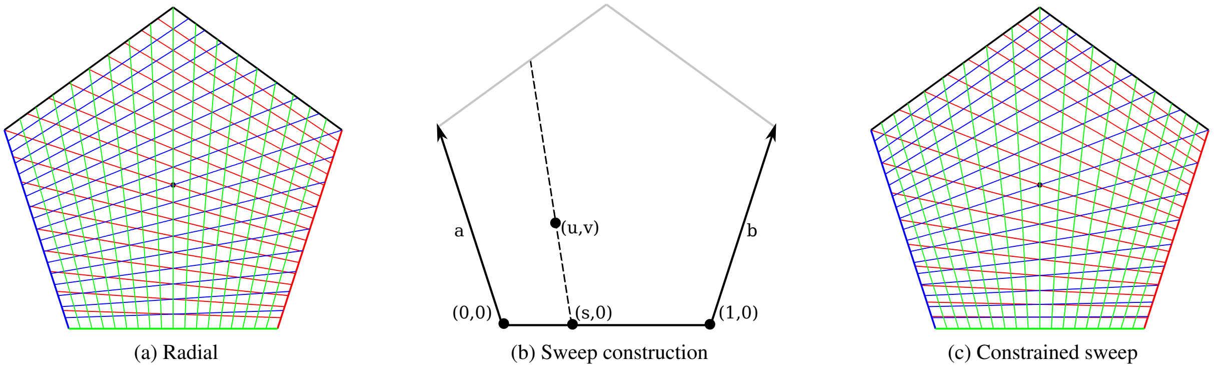
which leads to the depressed cubic equation

$$2\alpha + 16\kappa^2\alpha^3 l^2 = 1,$$

that can be solved by Cardano's method.

4. Surface Creation

There are various transfinite surface schemes that can be applied to interpolate given ribbons. For a comparison, see a review of the authors[13]. Here we will use Gregory patches, as it is one of the simplest and most well-known methods.



(a) Radial (b) Sweep construction (c) Constrained sweep

Figure 4: Parameterizations

4.1. Ribbons

Gregory patches combine corner interpolants, so our first task is to convert our side-based (linear or parabolic) ribbons to corner-based surfaces. It is well-known from the classical theory of Boolean-sum surfaces[2] that a correction patch $Q_{i,i-1}$ is needed to cancel out the unwanted terms coming from the combination of the two side interpolants (see also Section 4.3):

$$\begin{aligned} R_{i,i-1}(s_i,s_{i-1}) &= R_{i-1}(s_{i-1},s_i)+R_i(s_i,1-s_{i-1}) \\ &- Q_{i,i-1}(s_i,s_{i-1}), \end{aligned}$$

where $s_j = s_j(u,v)$ ($j \in [1\dots n]$) denote the side parameters defined over the polygonal domain. This brings us to the next topic: how to map the four-sided ribbon surfaces onto the n-sided domain polygon, or inversely, how to determine the local side parameters from a given (u,v) point.

4.2. Parameterization

The domain of a Gregory patch is an n-sided polygon in the 2D plane. Previous research[13] shows that the use of irregular polygons reflecting the spatial distribution of the boundary curves generally improves the quality of transfinite surfaces. In this paper, however, we will use regular domains for simplicity's sake. Experience shows that these behave well unless we have extreme boundary configurations with very uneven side lengths or sudden curvature changes.

Let us determine the parameter s_i for the i-th side. Traditionally, Gregory patches are parameterized by radial sweeping lines[1] (Figure 4a), connecting the domain point in question to the intersection of the extended polygon sides $i-1$ and $i+1$. This is suitable for G^1 continuous surfaces, but further differential properties are required for G^2 continuity. For each point on the i-th side, it is a necessary condition that the parametric speed of the adjacent side parameters $s_{i-1}$and s_{i+1} is identical, i.e.,

$$\frac{\partial s_{i-1}}{\partial w} = \frac{\partial s_{i+1}}{\partial w},$$

where w is an arbitrary sweeping direction. (See also the proof of continuity given in the Appendix.)

The above property can be satisfied, if we create the sweeping lines using Hermite polynomials. Without loss of generality, let the base edge be a segment from $(0,0)$ to $(1,0)$, a and b edge vectors associated with sides $i-1$ and $i+1$, respectively, and (u,v) the point to be mapped, see Figure 4b. Then we can construct the equation

$$(u,v) = (s,0) + d\,[a \cdot H(s) + b \cdot H(1-s)],$$

where $H(s) = 2s^3 - 3s^2 + 1$, and s and d are unknown. This leads to a fourth-degree equation in s:

$$c_4 s^4 + c_3 s^3 + c_2 s^2 + c_1 s + c_0 = 0,$$

where the coefficients are

$$\begin{aligned} c_4 &= \frac{2}{v}(a^v - b^v), \\ c_3 &= 2(a^u - b^u) + \frac{1}{v}(2u+3)(b^v - a^v), \\ c_2 &= 3(b^u - a^u) - \frac{u}{v}3(b^v - a^v), \\ c_1 &= \frac{1}{v}a^v, \\ c_0 &= a^u - \frac{u}{v}a^v, \end{aligned}$$

with $a = (a^u, a^v)$ and $b = (b^u, b^v)$.

This does not pose any difficulty for real-time computation, as efficient algorithms exist for solving fourth-degree polynomial equations[11], and the values for a given resolution can be cached. For the result, see Figure 4c. Note, that now the blue sweeplines of side $i-1$ and the red sweeplines of side $i+1$ at the bottom are identical in a differential sense.

4.3. Correction Terms

Let us investigate the partial derivatives of two ribbons meeting at a common corner point. We need a single corner inter-

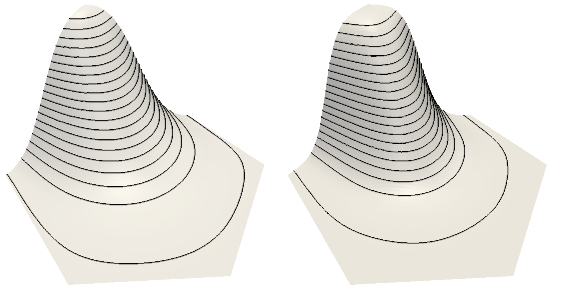


Figure 5: Blend function with $m = 2$ (left) and $m = 3$ (right)

polant, but the partial derivatives are not necessarily identical, which may pose a problem called *twist incompatibility*. Let us introduce the notation $t_i = 1 - s_{i-1}$. We would need

$$\begin{aligned}\frac{\partial^{p+q}}{\partial s_i^p \partial t_i^q} R_i(0,0) &= \frac{\partial^{p+q}}{\partial t_i^q \partial s_i^p} R_{i-1}(1,0) \\ &=: W_{p,q} \quad p,q \in \{0,1,2\}.\end{aligned}$$

Every curve network satisfies this equation for $p = q = 0$, as the boundaries meet at a fixed corner point. It is also natural to require that the equation holds for $p = 0$, $q = 1$ and $p = 1$, $q = 0$, constraining the first cross-derivative at the boundary to the tangent of the neighboring curve. Most networks do not go beyond this point — and even if the curves match a common surface curvature[5], it only handles the additional cases $p = 2$, $q = 0$ and $p = 0$, $q = 2$, as well as $p = q = 1$.

When some partial derivatives of the ribbons are not compatible, we need to apply Gregory's rational twists. These replace the constant vectors $W_{p,q}$ by rational expressions combining the two parametric variables (see below). In our current research, we assume that the boundary curves match only in position, and for all other terms rational expressions are used. This may create another layer of flexibility for shape optimization.

The correction patch is defined as

$$\begin{aligned}Q_{i,i-1}(s_i,t_i) &= P_i(0) + s_i W_{1,0} + t_i W_{0,1} + s_i t_i W_{1,1} \\ &+ \frac{1}{2} s_i^2 W_{2,0} + \frac{1}{2} t_i^2 W_{0,2} + \frac{1}{2} s_i^2 t_i W_{2,1} \\ &+ \frac{1}{2} s_i t_i^2 W_{1,2} + \frac{1}{4} s_i^2 t_i^2 W_{2,2},\end{aligned}$$

where each W is a rational function of s_i and t_i. The computation of these is a fairly straightforward generalization of the classical Gregory twists[15]. As an illustration, we show two such terms, the remaining ones are similar:

$$\begin{aligned}W_{1,0}(s_i,t_i) &= \frac{s_i^2 T_{i-1}(1) + t_i^3 \frac{\partial}{\partial s_i} P_i(0)}{s_i^2 + t_i^3}, \\ W_{1,2}(s_i,t_i) &= \frac{s_i^2 \frac{\partial^2}{\partial t_i^2} T_{i-1}(1) + t_i \frac{\partial}{\partial s_i} C_i(0)}{s_i^2 + t_i}.\end{aligned}$$

Substituting 0 for either s_i or t_i eliminates one of the conflicting partial derivatives.

4.4. Blending Functions

Every transfinite surface scheme combines individual interpolants by special blending functions that ensure the required interpolation properties and gradually vanish as we move towards the center of the domain. Gregory patches consist of corner interpolants, so we need *corner blend*s that interpolate the corner points, gradually fade on the adjacent sides, and vanish on all other sides.

For each (u,v) point in the polygonal domain, we determine an n-tuple of distance values Δ_i, computed as the perpendicular distances from the i-th side. Let $D_{i_1 \dots i_k} = \prod_{j \notin \{i_1 \dots i_k\}} \Delta_j^m$, then the corner blend is defined as

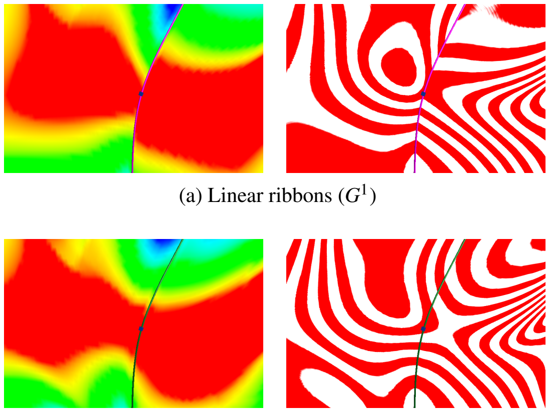
$$B_{i,i-1}(u,v) = \frac{D_{i,i-1}}{\sum_j D_{j,j-1}} \left(= \frac{1/(\Delta_i \Delta_{i-1})^m}{\sum_j 1/(\Delta_j \Delta_{j-1})^m} \right).$$

This function satisfies all requirements — it yields 1 at the $(i-1,i)$ corner, ensures a "gradual" $1 \to 0$ transition on sides $i-1$ and i as we move away from the corner, and vanishes on all the remaining sides.

The exponent m controls how rapidly the contribution of a single ribbon changes, compare the two images in Figure 5. It also ensures that the resulting surface retains the $(m-1)$-th derivatives of the ribbons, so we can use $m = 2$ for linear ribbons, and $m = 3$ for parabolic ribbons to achieve G^1 and G^2 continuity, respectively.

4.5. Surface Equation

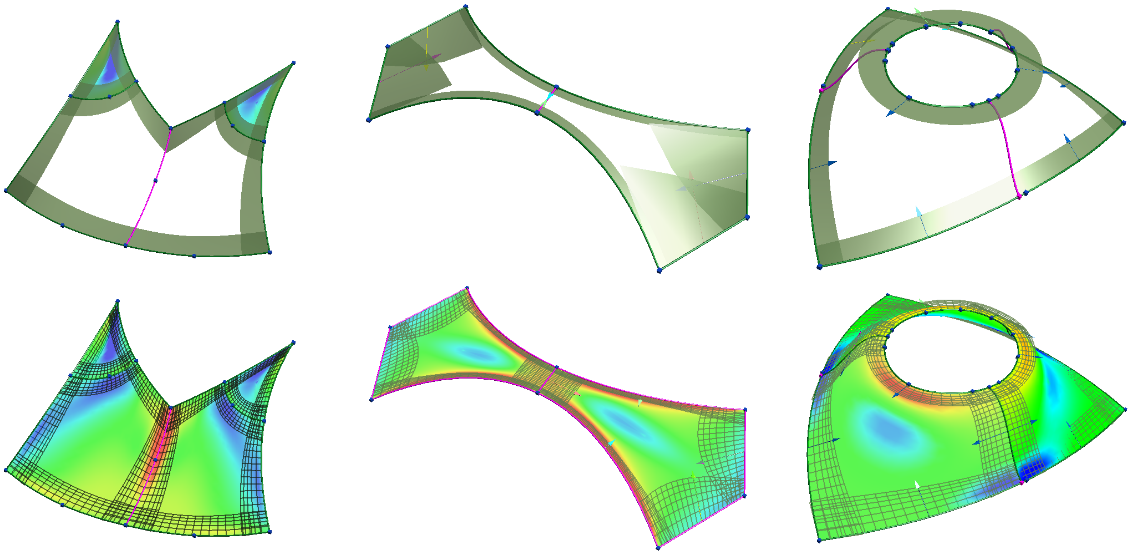
Given a curve network with side interpolants, now we can create all constituents — the corner interpolants, the domain



(a) Linear ribbons (G^1)

(b) Parabolic ribbons (G^2)

Figure 6: Connectivity between Gregory patches



(a) Concave angle (b) Highly curved boundary (c) Internal loop

Figure 7: Improved surfacing using connection curves.

polygon, the parameterization and the blending functions. Putting these together, we arrive at

$$S(u,v) = \sum_{i=1}^{n} R_{i,i-1}(s_i(u,v), s_{i-1}(u,v)) B_{i,i-1}(u,v).$$

5. Examples

Figure 6 shows two adjacent patches with linear vs. parabolic ribbons. In this example, the target curvatures were computed by averaging those on the left and right sides. It can be seen, that with parabolic ribbons the curvatures nicely match, and the isophote strips smoothly change across the shared boundary, showing G^2 continuity.

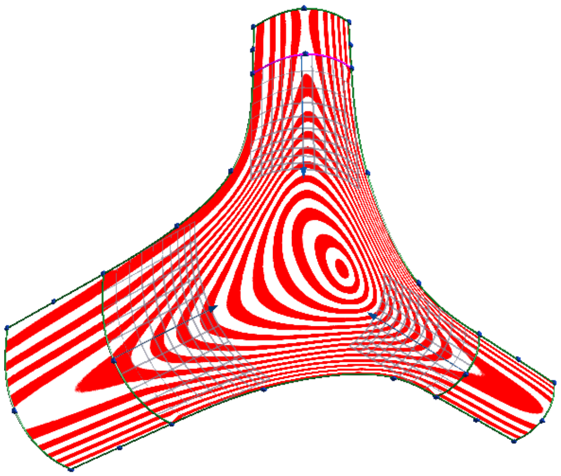


Figure 8: Vertex blend with three master surfaces

In the rest of this section, we present some applications of parabolic ribbons.

5.1. Difficult Curve Network Configurations

While most of the time G^1 transfinite surface interpolation produces a collection of smoothly connected convex patches, there are certain cases, where in order to handle complex configurations or avoid shape artifacts, we need to insert "artificial" *connection curves* into the network:

- curve loops with a concave angle,
- avoiding distortions due to highly curved boundaries,
- connect internal loops (holes).

Examples are shown in Figure 7. In these cases, first we insert connection curves, then create G^1 patches. After taking the average of the curvatures, we compute parabolic ribbons and regenerate the patches now with curvature continuity. These composite patches remain smooth internally and the seamlines are invisible along the connection curves. (Note, that generally connection curves are automatically generated, and remain hidden from the users.)

5.2. Master-Slave constructions

There are various methods to compute a target curvature function along a given curve. The most straightforward solution is averaging the curvatures of two adjacent G^1 patches,

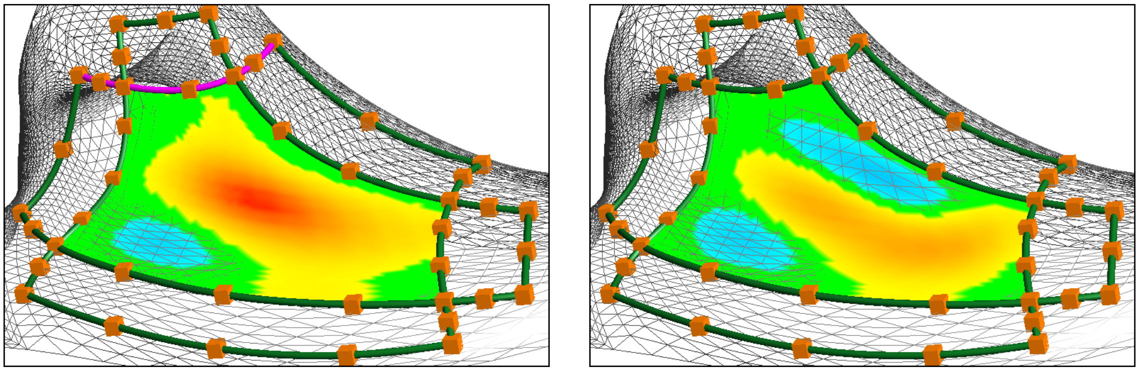


Figure 9: Deviation from the mesh using linear (left) and parabolic (right) ribbons

as before. Another typical situation is, when the curvature of a *master* patch needs to be retained. Then these curvatures are propagated to the surrounding *slave* patches using parabolic ribbons. Such an example is shown in Figure 8, where a setback vertex blend was created, satisfying curvature continuity joining three edge blends, i.e., we have three master surfaces, and one slave in the middle.

5.3. Mesh Approximation

Creating a concise representation of a mesh is an important task in general topology surface modeling. First, a network of curves is drawn on the mesh, which also defines a multi-sided patch structure. By deducing boundary curves and cross-derivatives from the mesh, transfinite surfaces can nicely approximate the data points in the interior. Using locally estimated normal vectors we can create normal fences and linear ribbons for G^1 patches. If we estimate local curvatures along the boundaries, as well, this makes it possible to create parabolic ribbons and G^2 patches, which will produce smoother and more accurate surface models (see Figure 9).

Conclusion

We have discussed an approach for G^2 transfinite surface interpolation combining parabolic ribbons. These ribbons match curvatures that are associated with the edges of a general topology curve network. The formulation is based on corner interpolants and a special sweepline parameterization. Continuity issues and the computation of linear and parabolic ribbons were explained in details. A few applications where G^2 ribbons are needed have also been presented.

There are several open issues in transfinite surface interpolation for future research. Ribbon creation is one of the fundamental ones, as they are constrained, but not uniquely defined. We are currently investigating ribbon optimization approaches for surface fairing and obtaining the best possible transfinite approximations over triangular meshes.

Acknowledgements

This work was supported by the Hungarian Scientific Research Fund (OTKA, No. 101845). The pictures in this paper were generated by the Sketches system developed by ShapEx Ltd., Budapest. The contribution of György Karikó to develop this prototype system is highly appreciated.

References

1. P. Charrot, J. A. Gregory, A pentagonal surface patch for computer aided geometric design, Computer Aided Geometric Design 1 (1) (1984) 87–94.
2. S. A. Coons, Surfaces for computer-aided design of space forms, Tech. Rep. MIT/LCS/TR-41, Massachusetts Institute of Technology (1967).
3. J. A. Gregory, J. M. Hahn, A C^2 polygonal surface patch, Computer Aided Geometric Design 6 (1) (1989) 69–75.
4. R. Hall, G. Mullineux, Continuity between gregory-like patches, Computer aided geometric design 16 (3) (1999) 197–216.
5. T. Hermann, G. Lukács, F.-E. Wolter, Geometrical criteria on the higher order smoothness of composite surfaces, Computer Aided Geometric Design 16 (9) (1999) 907–911.
6. K. Hormann, M. S. Floater, Mean value coordinates for arbitrary planar polygons, Transactions on Graphics 25 (4) (2006) 1424–1441.
7. K. Kato, Generation of n-sided surface patches with holes, Computer-Aided Design 23 (10) (1991) 676–683.
8. K. Kato, N-sided surface generation from arbitrary boundary edges, in: Curve and Surface Design, Innovations in Applied Mathematics, Vanderbilt University Press, 2000, pp. 173–181.

9. H. P. Moreton, C. H. Sequin, Minimum variation curves and surfaces for computer-aided geometric design, in: N. S. Sapidis (ed.), Designing Fair Curves and Surfaces, SIAM, 1994, pp. 123–159.

10. J. Pegna, F.-E. Wolter, Geometrical criteria to guarantee curvature continuity of blend surfaces, Journal of Mechanical Design 114 (1) (1992) 201–210.

11. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical recipes in C (2nd ed.): the art of scientific computing, Cambridge University Press, 1992.

12. P. Salvi, T. Várady, A. Rockwood, Ribbon-based transfinite surfaces, Computer Aided Geometric Design (submitted).

13. T. Várady, A. Rockwood, P. Salvi, Transfinite surface interpolation over irregular n-sided domains, Computer Aided Design 43 (11) (2011) 1330–1340.

14. W. Wang, B. Jüttler, D. Zheng, Y. Liu, Computation of rotation minimizing frames, Transactions on Graphics 27 (1) (2008) 2.

15. A. Worsey, A modified C2 Coons' patch, Computer Aided Geometric Design 1 (4) (1984) 357–360.

Appendix A: Proof of Continuity

We will provide here a short proof that Gregory patches with matching parabolic ribbons are indeed G^2 continuous. We will prove a stronger statement: the modified Gregory patches described in the paper interpolate their ribbons with C^2 continuity.

To save space, arguments of functions are omitted, when this cannot cause any misunderstanding. From here onwards, we look at a point on the ith side, i.e., $s_{i-1} = 1$ and $s_{i+1} = 0$. In this situation, the blending function has several important properties[12], which we list here without proof:

$$\begin{aligned}
B_{i,i-1} + B_{i+1,i} &= 1, && (1)\\
\frac{\partial}{\partial w} B_{j,j-1} &= 0, \quad j \notin \{i, i+1\}, && (2)\\
\frac{\partial^2}{\partial w^2} B_{j,j-1} &= 0, \quad j \notin \{i, i+1\}, && (3)
\end{aligned}$$

w being an arbitrary direction. From the above it also follows that

$$\begin{aligned}
\frac{\partial}{\partial w}(B_{j,j-1} + B_{j+1,j}) &= 0, \quad j \notin \{i-1, i+1\}, && (4)\\
\frac{\partial^2}{\partial w^2}(B_{j,j-1} + B_{j+1,j}) &= 0, \quad j \notin \{i-1, i+1\}. && (5)
\end{aligned}$$

C^0 Continuity

This is very straightforward, using property (1) of the blend functions:

$$\begin{aligned}
S &= R_{i,i-1} B_{i,i-1} + R_{i+1,i} B_{i+1,i}\\
&= R_i(s_i, 0)(B_{i,i-1} + B_{i+1,i}) = R_i(s_i, 0).
\end{aligned}$$

C^1 Continuity

As before, most of the equation vanishes, leaving

$$\begin{aligned}
\frac{\partial}{\partial w} S &= \left[\frac{\partial}{\partial s_{i+1}} R_{i+1,i} \frac{\partial s_{i+1}}{\partial w} + \frac{\partial}{\partial s_i} R_{i+1,i} \frac{\partial s_i}{\partial w}\right] B_{i+1,i}\\
&+ \left[\frac{\partial}{\partial s_i} R_{i,i-1} \frac{\partial s_i}{\partial w} + \frac{\partial}{\partial s_{i-1}} R_{i,i-1} \frac{\partial s_{i-1}}{\partial w}\right] B_{i,i-1}\\
&+ \left[R_{i+1,i} \frac{\partial}{\partial w} B_{i+1,i} + R_{i,i-1} \frac{\partial}{\partial w} B_{i,i-1}\right].
\end{aligned}$$

Since $R_{i+1,i} = R_{i,i-1}$, we can use property (4) to eliminate the last term. After some calculation, we arrive at

$$\frac{\partial}{\partial w} S = P_i'(s_i) \frac{\partial s_i}{\partial w} + T_i(s_i) \left[B_{i+1,i} \frac{\partial s_{i+1}}{\partial w} + B_{i,i-1} \frac{\partial s_{i-1}}{\partial w}\right],$$

which is a combination of the ribbon's side- and cross-derivatives, proving G^1 continuity. Using the parameterization constraint $\frac{\partial s_{i+1}}{\partial w} = \frac{\partial s_{i-1}}{\partial w}$, we get back the ribbon's first derivative, so we have proved C^1 continuity.

C^2 Continuity

In a similar vein, we can eliminate a large part of the equation, after which the following remains:

$$\begin{aligned}
\frac{\partial^2}{\partial w} S &= 2\left[\frac{\partial}{\partial s_{i+1}} R_{i+1,i} \frac{\partial s_{i+1}}{\partial w} + \frac{\partial}{\partial s_i} R_{i+1,i} \frac{\partial s_i}{\partial w}\right] \frac{\partial}{\partial w} B_{i+1,i}\\
&+ 2\left[\frac{\partial}{\partial s_i} R_{i,i-1} \frac{\partial s_i}{\partial w} + \frac{\partial}{\partial s_{i-1}} R_{i,i-1} \frac{\partial s_{i-1}}{\partial w}\right] \frac{\partial}{\partial w} B_{i,i-1}\\
&+ \left[\frac{\partial^2}{\partial s_{i+1}^2} R_{i+1,i} \frac{\partial s_{i+1}^2}{\partial w^2} + 2 \frac{\partial^2}{\partial s_{i+1} s_i} R_{i+1,i} \frac{\partial s_i}{\partial w} \frac{\partial s_{i+1}}{\partial w}\right.\\
&+ \left.\frac{\partial^2}{\partial s_i^2} R_{i+1,i} \frac{\partial s_i^2}{\partial w^2}\right] B_{i+1,i} + \left[\frac{\partial^2}{\partial s_i^2} R_{i,i-1} \frac{\partial s_i^2}{\partial w^2}\right.\\
&+ \left. 2 \frac{\partial^2}{\partial s_i s_{i-1}} R_{i,i-1} \frac{\partial s_{i-1}}{\partial w} \frac{\partial s_i}{\partial w} + \frac{\partial^2}{\partial s_{i-1}^2} R_{i,i-1} \frac{\partial s_{i-1}^2}{\partial w^2}\right] B_{i,i-1}\\
&+ \left[R_{i+1,i} \frac{\partial^2}{\partial w^2} B_{i+1,i} + R_{i,i-1} \frac{\partial^2}{\partial w^2} B_{i,i-1}\right].
\end{aligned}$$

The last term vanishes once again, due to property (5). This leads to

$$\begin{aligned}
\frac{\partial^2}{\partial w^2} S &= 2T_i(s_i) \left[\frac{\partial}{\partial w} B_{i+1,i} \frac{\partial s_{i+1}}{\partial w} + \frac{\partial}{\partial w} B_{i,i-1} \frac{\partial s_{i-1}}{\partial w}\right]\\
&+ P_i''(s_i) \frac{\partial s_i^2}{\partial w^2} + 2T_i'(s_i) \frac{\partial s_i}{\partial w} \left[B_{i+1,i} \frac{\partial s_{i+1}}{\partial w} + B_{i,i-1} \frac{\partial s_{i-1}}{\partial w}\right]\\
&+ C_i(s_i) \left[B_{i+1,i} \frac{\partial s_{i+1}^2}{\partial w^2} + B_{i,i-1} \frac{\partial s_{i-1}^2}{\partial w^2}\right].
\end{aligned}$$

Applying the parameterization constraint, we get

$$\frac{\partial^2}{\partial w^2} S = P_i''(s_i) \frac{\partial s_i^2}{\partial w^2} + 2T_i'(s_i) \frac{\partial s_i}{\partial w} \frac{\partial d_i}{\partial w} + C_i(s_i) \frac{\partial d_i^2}{\partial w^2},$$

which is the second derivative of the ribbon, so we have proved C^2 continuity.