Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Constrained modeling of multi-sided patches

Péter Salvi*, Márton Vaitkus, Tamás Várady

Budapest University of Technology and Economics

ARTICLE INFO

Article history:
Received May 31, 2023

Keywords: Multi-sided Surfaces, Curved Domain, Constrained Editing, Control Vectors, Geometric Continuity

ABSTRACT

We investigate genuinely multi-sided patches that interpolate ribbon surfaces along their boundaries. Recent works suggest defining patches over parametric domains with curved boundaries and hole loops, where the domain mimics the shape of the surface to be constructed. Cross-derivatives of the input are interpreted with respect to this curved parametric domain, but it is an open question how to initialize and modify these vector functions.

We propose algorithms to set the cross-derivatives of multi-sided patches defined by Bézier and B-spline ribbons. Boundaries and surface constraints are inherited from adjacent patches, and our goal is to define a nice surface while ensuring smooth (G^1) connections. We exploit that ribbon parameterizations induce 'proportional' cross-derivative magnitudes in 3D, and express cross-derivatives as the combination of vector functions and appropriately chosen scalar functions.

Continuity constraints imply complex relations between the control points of the ribbons, so their direct modification is not feasible. Instead we suggest a constrained editing technique based on control vectors that significantly simplifies this task.

1. Introduction

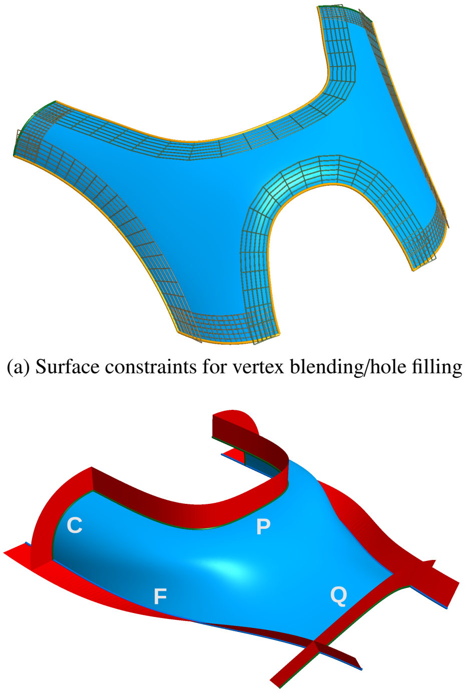
We investigate ribbon-based, genuine multi-sided surfaces satisfying positional and cross-derivative boundary constraints, thereby smoothly connecting to adjacent tensor-product and multi-sided surfaces, while also possessing a natural blend in the interior. Recent publications suggest the use of parametric domains with curved boundaries and hole loops, see the Generalized Bézier and B-spline patches in Várady et al. [1] and Vaitkus et al. [2], respectively. Curved domains are inevitable in cases where the boundary curves are placed irregularly, have significant variation in length, and contain strongly concave segments (this is illustrated later in Fig. 12a).

The majority of publications on multi-sided patch representations in the CAGD literature deals with surfacing schemes where predefined surface interpolants (*ribbons*, see e.g. Salvi et al. [3]) are combined using various blending functions. In practice, however, such ribbons are not given *a priori*, and their construction, while an important issue, seems to be a somewhat unexplored area. Our interest is to define genuine multi-sided patches based on tangential surface constraints, exploiting only partially given cross-derivative information. We deal with two typical scenarios:

1. For hole filling or vertex blending the surrounding surfaces explicitly specify constraints for cross-derivatives (G^1), see e.g. Figure 1a.
2. In curve network based design the cross-derivatives are supposed to approximately (NG^1) match smooth vector functions shared across the boundaries, such as rotation minimizing frames; an example is shown in Figure 1b. (We use the notation NG^k for numerical/approximative G^k-continuity, see also Section 4.1.)

It is well-known that the first derivatives of high-quality para-

*Corresponding author:
e-mail: salvi@iit.bme.hu (Péter Salvi)



(a) Surface constraints for vertex blending/hole filling

(b) Normal fence constraints for curvenet-based design

Figure 1: Different applications of multi-sided surfaces.

metric curves and surfaces need to be set proportionally to the extent of the whole shape, as this determines the ‘velocity’ of these curves and surfaces at their parametric boundaries. The magnitude of the first derivatives cannot be inferred from a regular polygonal domain. However, if we have a curved domain that is a moderately distorted proportional map of the multi-sided patch, we can deduce information related to the magnitude of the individual ribbons. In fact, interpretation of parametric cross-derivatives in a consistent manner is possible only based on the (curved) domain and its parameterization. Larger or smaller magnitudes typically increase or decrease the curvature of the surface interior. In extreme cases, large cross-derivatives overshoot and the patch wrinkles, while small derivatives produce flat spots, see Fig. 13. Note that proportional parameterization is a well-accepted idea in CAGD, take for example non-uniform B-splines, where the knot vector and the placement of the control points need to be set according to the shape.

Setting suitable initial cross-derivatives for multi-sided patches is a difficult problem. When we deal with complex geometries and hole loops, the cross-derivative functions may have complex shapes. For each boundary curve the ‘opposite’ part of the domain will determine the magnitudes of the cross-derivatives, as these are in strong correspondence with the surface to be constructed. The following sequence illustrates our concept. Figure 2a shows a portion of a curve network with a given G^1 constraint inherited from the patch below. Figure 2b shows a ‘narrow’ 4-sided patch, where in order to avoid overshooting, the cross-derivative function must be pulled back. Figure 2c is an extended 6-sided patch, and an upscaled cross-derivative function is required to avoid flatness in the middle of the surface. Finally, in Figure 2d, a hole loop has been inserted, thus the related cross-derivative should be contracted accordingly.

Our contribution in this paper is twofold. We assume that the cross-derivative constraints from the surrounding patches are fixed. First we introduce an algorithm to construct proportional initial ribbons for multi-sided patches. Then we describe how to edit the interior control points of ribbons simultaneously, and thus how to adjust the magnitudes of the cross-derivative functions. In our project we deal with G^1 surface constraints, and apply these techniques primarily to the recently introduced Generalized Bézier (GB) and Generalized B-spline (GBS) patches, see Várady et al. [1] and Vaitkus et al. [2], respectively. It is possible to generalize the proposed technique for G^2 continuity, but the algebra is quite cumbersome and leads to high-degree ribbons. We believe that ensuring approximate curvature continuity produces surfaces with better curvature distribution—see also similar thoughts in Karčiauskas and Peters [4]. A particular solution for connecting GBS patches with NG^2 was reported in Vaitkus et al. [2].

In Section 2 we briefly review the related literature. In Section 3 we describe how cross-derivatives are constructed and show how weighting functions for scaling and shearing can be set by means of control vectors. In Section 4 we investigate algorithms for setting and editing constrained cross-derivatives. Finally, in Section 5 we evaluate the proposed method and show a few more test cases.

2. Previous work

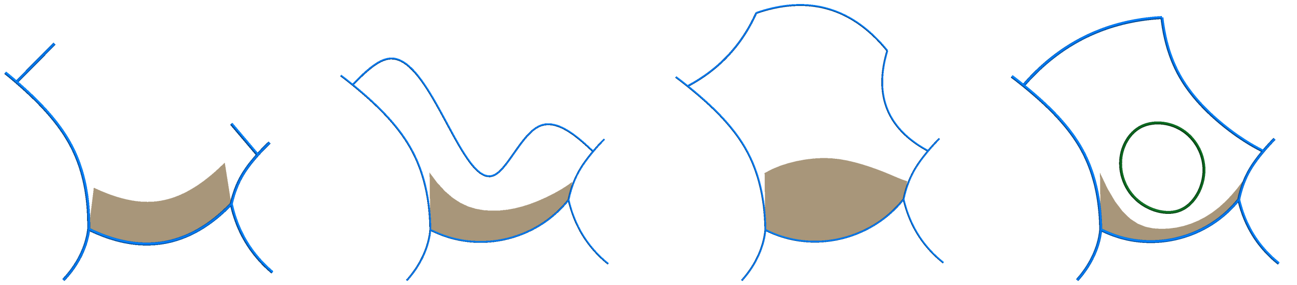
2.1. Multi-sided surfaces

We consider only genuinely multi-sided surfaces—see Malraison [5], Várady et al. [6]. We are aware of only a few surface representations that use curved domains (e.g. Martin and Reif [7], Sabin et al. [8]) and allow multiply-connected topology (e.g. Kato [9], Sabin [10]). Generalized Bézier [1] and Generalized B-Spline [2] patches possess these capabilities and thus form the foundation for our examples.

2.2. G^1-continuous splines

The *geometric continuity* of surfaces is a classical topic within CAGD—see Peters [11] or Kiciak [12]. We are primarily concerned with G^1 (i.e., tangent plane) continuity of Bézier and B-spline patches. The necessary-and-sufficient geometric conditions for G^1 continuity are highly nonlinear, so in practice *sufficient* conditions are considered that enforce a particular linear dependence between (cross-)derivatives (equivalent to C^1 continuity after a (fixed) reparameterization, see Peters [11]). We rely on a formulation of G^1 continuity that explicitly defines one of the cross-derivatives as a combination of the derivative of the common curve and the cross-derivative of a surface constraint [13]; this form is equivalent to the most general G^1 constraint [14] and is often preferred in practice (e.g. Hoschek and Lasser [15], Renner [16], Shi et al. [17]).

To gain more flexibility, B-splines are often used instead of high-degree polynomials, but G^1 continuity poses some difficulties [18]. In particular, industry-standard degree-3 splines



(a) Portion of a curve network and a constraining surface

(b) Opposite boundary narrows the cross-derivative

(c) Opposite boundaries enlarge the cross-derivative

(d) Hole in the middle shrinks the cross-derivative

Figure 2: Cross-derivatives adapt to various boundary configurations.

cannot be G^1-continuous across arbitrary curves, unless their regularity is at most C^1 [19], or their degree is raised to at least 5 [20]. As a consequence, cubic B-spline surfaces with NG^1 continuity are often preferred, as long as the error is kept within a prescribed tolerance [21]; see Kiciak [12] for relevant results of approximation theory.

2.3. Automatic setting of derivatives

Adaptive data parameterization is an important aspect of spline curve fitting and interpolation, where chordal and centripetal methods are popular choices with good theoretical properties [22]. Extending non-uniform parameterizations to non-tensor-product surfaces is a challenging problem due to difficulties of defining a suitable parametric domain [23], so a procedural approach is used in practice, where patches are individually defined with G^k continuity constraints. Shirman and Séquin [24] carried out an in-depth empirical study of possible heuristics for setting derivatives of G^1 Hermite curves. The majority of existing geometrically continuous spline constructions consider exclusively topological information [25], or operate over a fixed multi-patch parameterization [26].

Cross-derivatives in a general topology network are often constructed from boundary curves by interpolating derivatives determined at the endpoints [27, 17], but this does not take the global shape of the patch into account. Konno et al. [28] presented heuristics for determining derivative magnitudes of 4-sided ribbon-based patches. Kiciak [12] describes various methods for setting cross-derivatives of tensor-product surfaces, based on 3D distances between opposing curves to be interpolated, as well as mapping vector sweeps along 2D trimming curves into 3D. Shen et al. [29] used a local parameterization of a curved domain to fix the derivatives of a non-uniform subdivision surface approximating a trimmed NURBS. The previous two methods directly set cross-derivative vectors, while our approach, in contrast, produces scalar functions for general G^1-continuous connections. We are not aware of any work that studied the problem of setting cross-derivatives for genuinely multi-sided patches.

2.4. Constrained editing

In a seminal work, Shirman and Séquin [30] investigated the possibilities for editing Bézier ribbon surfaces of Gregory patches while preserving G^1 continuity. For cubic ribbons, the authors identified degrees of freedom allowed by continuity constraints—named *bulge*, *shear*, and *tilt*—that can be edited in an intuitive manner via the central coefficient of a common quadratic surface constraint. Konno, Chiyokura and others later extended this idea to higher order Gregory–Coons patches [28], as well as B-spline and NURBS boundaries [31, 32]. In these works, a notion of *control vector* was introduced, which referred to vector coefficients of a surface constraint shared by adjacent surfaces. B-spline ribbons were converted into C^0 piecewise polynomials, apparently to avoid dealing with the complexities of general G^1 B-spline constraints. In contrast, our proposed constrained editing framework handles Bézier and B-spline ribbons with general degree and knot structure, avoiding algebraic difficulties that hindered previous attempts.

3. Constrained cross-derivatives

In this section we describe our cross-derivative concept in a fairly general setting, but our algorithms will be introduced in Section 4 by ribbons given in Bézier and B-spline form. The test surfaces in this paper were produced using Generalized Bézier (GB) and Generalized B-spline (GBS) patches; their formulation is briefly presented in Appendix A. Note, however, that our apparatus is also applicable to the constrained editing of general G^1 ribbon-based surfaces.

3.1. A general multi-sided surface model

We investigate interpolating multi-sided patches, defined by a combination of open or periodic ribbon surfaces that prescribe positions and cross-derivatives along the surface boundaries. The surfaces, denoted by $\mathbf{S}(u, v)$, are defined over a *curved domain* in the (u, v) plane, which may have curved boundaries and hole loops. A fundamental assumption here is that the domain resembles the surface to be constructed, i.e., it is a 'flattened' version of the surface with moderate distortion. One particular option to produce curved domains is detailed in Várady et al. [1], where (in essence) a *development* of the boundary curves is computed in the tangent planes of the ribbon interpolants, and thus the shape of the boundary is preserved as seen from 'within the surface'. More technically, geodesic curvature is preserved as much as possible. Alternative curve domain algorithms can also be derived by flattening a coarse preliminary 3D triangulation, see for example Zou et al. [33].

For each ribbon $\mathbf{R}_i(s_i, h_i)$ there is a parameterization that assigns local (s_i, h_i) parameters for all (u, v) points of the domain. Parameterization determines how the points of the ribbon are mapped onto the multi-sided patch. The side parameter $s_i = s_i(u, v)$ varies linearly on side i between 0 and 1; the distance parameter $h_i = h_i(u, v)$ vanishes on side i and increases monotonically within the domain, eventually reaching 1 on the *distant* sides $(j \neq i-1, i, i+1)$.

Parameterizing ribbons is a complex problem, in particular for curved domains. While there is an intense search for alternative methods, in our project we use *harmonic functions*, which are constrained minimizers of the Dirichlet energy

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \int_D |\nabla f|^2 \, dA \\ \text{subject to} \quad & f(x) = b_f(x), x \in \mathcal{D}_f. \end{aligned} \tag{1}$$

Here $f = s_i$ or h_i, as defined above, and b_f encodes the Dirichlet boundary conditions for the constrained subset of the domain boundary $\mathcal{D}_f$. Harmonic functions are preferred, as they are C^∞-continuous in the interior, and take their minimal and maximal values on the boundary. For hole loops a periodic parameterization is used, following Vaitkus et al. [2].

3.2. Cross-derivatives with constraints

For each ribbon $\mathbf{R}_i(s_i, h_i)$, we construct a cross-derivative function, $\mathbf{R}_{h,i}(s_i, 0) = \frac{\partial}{\partial h_i}\mathbf{R}_i(s_i, 0)$, that must satisfy external G^1 surface constraints in the form of vectors functions $\mathbf{D}_i(s_i)$ representing cross-derivatives from the adjacent patches. We apply the following well-known formula [15, Chapter 7] (index i and parameter h are omitted from now on for convenience):

$$\mathbf{R}_h(s) = \alpha(s)\mathbf{D}(s) + \beta(s)\mathbf{R}_s(s). \tag{2}$$

Here $\mathbf{R}_s(s)$ denotes the first derivative in the longitudinal direction, and $\alpha(s)$, $\beta(s)$ are scalar weighting functions: $\alpha(s)$ is responsible for *scaling*, i.e., setting the magnitude of $\mathbf{R}_h(s)$, while $\beta(s)$ provides *shearing*, i.e., the rotation of $\mathbf{R}_h(s)$ around the surface normal. This formula guarantees G^1 continuity, as at each point of the boundary the normal vector of the patch will be parallel to that of the constraining surface, i.e., $\mathbf{R}_s(s) \times \mathbf{R}_h(s) \parallel \mathbf{R}_s(s) \times \mathbf{D}(s)$. Obviously, in this type of construction the degree of the ribbons is generally raised.

There is a great variety how these functions can be chosen and naturally infinitely many cross-derivatives exist that match given surface constraints. Our goal is to define scalar functions with geometric meaning that can produce good initial settings and facilitate the editing of ribbons. The scalar functions possess certain degrees of freedom $d_{\alpha\beta}$; more DoFs permit more flexible cross-derivative functions. In our project we follow some simple—though not necessary—conventions:

1. For Bézier ribbons both scalar functions are given in Bézier form, and their degrees are assumed to be equal.
2. For B-spline ribbons, both scalar functions are set to cubic B-splines, defined over the same knot vector as the given boundary curve.

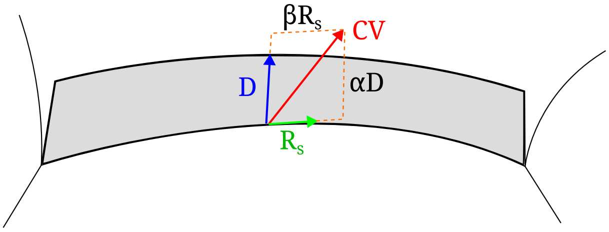


Figure 3: Components of a control vector.

In both cases, $d_{\alpha\beta}$ is determined by the related degrees and the number of knots.

The degrees of the prescribed vector functions $\mathbf{D}(s)$ and $\mathbf{R}_s(s)$ may differ, and we remark that the constraining surface often has a lower degree than the corresponding boundary. For example, $\mathbf{D}(s)$ can be a constant vector or a linear sweep, when the multi-sided patch joins simple surfaces, such as fillets or lofts. In other cases, $\mathbf{D}(s)$ may represent a planar vector function. As an example, take the simple test object in Figure 1b, defined by six cubic B-spline boundaries and two constant (C), one planar (P), one quadratic (Q) and two general free-form (F) constraint surfaces.

3.3. Control vectors

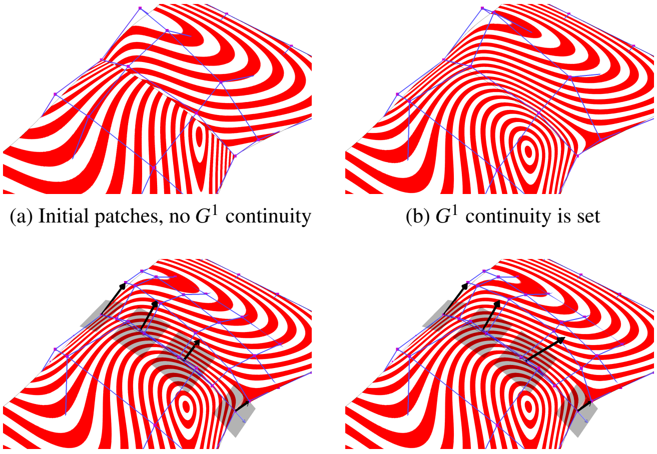
Take the cross-derivative function at an arbitrary parameter value s_k. The vector $\mathbf{R}_h(s_k)$ lies in the tangent plane, spanned by two local vectors $\mathbf{D}(s_k)$ and $\mathbf{R}_s(s_k)$, with coefficients $\alpha(s_k)$ and $\beta(s_k)$. Going the other way around: assume we want to adjust the cross-derivative function via a prescribed *control vector* $\mathbf{CV}_k$ at parameter s_k. The control vector unambiguously defines two scalar values α_k and β_k in the basis $(\mathbf{D}(s_k), \mathbf{R}_s(s_k))$ of the local tangent plane, see Fig. 3. If $\alpha(s_k) = \alpha_k$ and $\beta(s_k) = \beta_k$ are enforced, the related scalar functions will guarantee that $\mathbf{R}_h(s_k) = \mathbf{CV}_k$.

We may have a collection of constraints at various parameter values that will indirectly define the related α and β scalar functions at certain parameter values. As we modify the $\mathbf{CV}_k$-s, they change the corresponding cross-derivative function and the related control points. Note that relocating the control points of Bézier or B-spline ribbons directly in 3D, while retaining continuity, is a hopeless task in general, as the interdependency of the related control points is quite complex. However, if we perform editing by means of the control vectors, G^1 continuity is automatically guaranteed and the control points can be recalculated in a straightforward manner.

At this point we do not wish to limit the number or placement of the control vectors, and remark that they can be positioned at arbitrary parameter values, or by some rule such as uniform division, knot points or Greville abscissae, depending on the editing operation we wish to perform (see examples later).

3.4. A simple example

The above concept is demonstrated by a very simple example. Assume that our ribbon $\mathbf{R}(s, h)$ has a cubic boundary and a quadratic constraint function $\mathbf{D}(s)$. Thus the first derivative



(a) Initial patches, no G^1 continuity

(b) G^1 continuity is set

(c) Control vectors added

(d) Control vector 2 modified

Figure 4: Modification with control vectors, showing isophotes.

is also quadratic by s. All three vector functions are given in Bézier form:

$$\mathbf{R}_h(s) = \sum_{i=0}^{3} \mathbf{Q}_{h,i} B_i^3(s), \tag{3}$$

$$\mathbf{D}(s) = \sum_{i=0}^{2} \mathbf{D}_i B_i^2(s), \tag{4}$$

$$\mathbf{R}_s(s) = \sum_{i=0}^{2} \mathbf{Q}_{s,i} B_i^2(s). \tag{5}$$

In order to match the tangent vectors of the adjacent boundary curves at the corners, we set end constraints at $s = 0$ and $s = 1$; these can be satisfied by linear $\alpha(s)$ and $\beta(s)$ functions ($d_{\alpha\beta} = 2$):

$$\alpha(s) = \alpha_0(1 - s) + \alpha_1 s, \qquad \beta(s) = \beta_0(1 - s) + \beta_1 s. \tag{6}$$

The two control vectors $\mathbf{CV}_0$ and $\mathbf{CV}_1$ are fixed indirectly at the two ends:

$$\mathbf{R}_h(0) = \mathbf{CV}_0 = \alpha_0 \mathbf{D}_0 + \beta_0 \mathbf{Q}_{s,0}, \tag{7}$$

$$\mathbf{R}_h(1) = \mathbf{CV}_1 = \alpha_1 \mathbf{D}_2 + \beta_1 \mathbf{Q}_{s,2}. \tag{8}$$

Now assume that we wish to adjust the middle part of the cross-derivative function. New degrees of freedom are obtained by degree elevating to a new cubic function ($d_{\alpha\beta} = 4$) with Bernstein coefficients $\{\alpha_{0,}\alpha_{1,}\alpha_{2,}\alpha_3\}$ and $\{\beta_{0,}\beta_{1,}\beta_{2,}\beta_3\}$, where α_1, α_2 and β_1, β_2 are free to set. We may modify a related control vector $\mathbf{CV}_2$ at $s_k = \frac{2}{3}$, so that $\mathbf{CV}_2 = \alpha_2 \mathbf{D}(\frac{2}{3}) + \beta_2 \mathbf{R}_s(\frac{2}{3})$. This determines the weighting functions, and we can compute the related control points of the ribbon in a straightforward manner. A quadratic constraint and cubic weighting functions will lead to a quintic cross-derivative. In our example, if we perturb $\mathbf{CV}_2$ by

$$\alpha_2 \Rightarrow \alpha_2 + \Delta\alpha_2, \qquad \beta_2 \Rightarrow \beta_2 + \Delta\beta_2, \tag{9}$$

then the displacement of the interior control points will depend only on $\Delta\alpha_2$ and $\Delta\beta_2$.

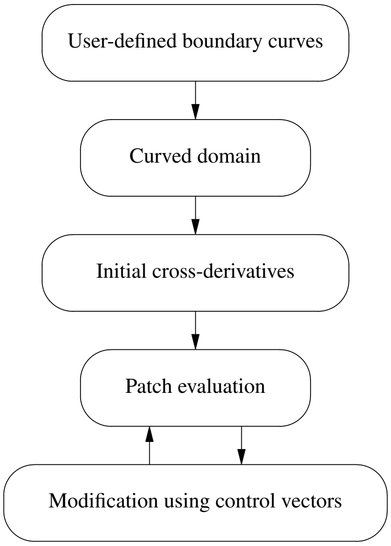

User-defined boundary curves
Curved domain
Initial cross-derivatives
Patch evaluation
Modification using control vectors

Figure 5: Overview of the general workflow.

Figure 4 illustrates this simple example. We start with two cubic patches without G^1 continuity. The bottom patch defines a quadratic cross-derivative constraint (Fig. 4a). Then by linear weighting functions we ensure G^1 continuity (Fig. 4b). Two new degrees of freedom are added by two control vectors in the middle producing a quintic patch (Fig. 4c). When the right control vector is modified, the interior control points are simultaneously relocated, and G^1 continuity is retained (Fig. 4d). These images with isophotes show perfect G^1 continuity (and also reasonably good numerical curvature continuity in this case).

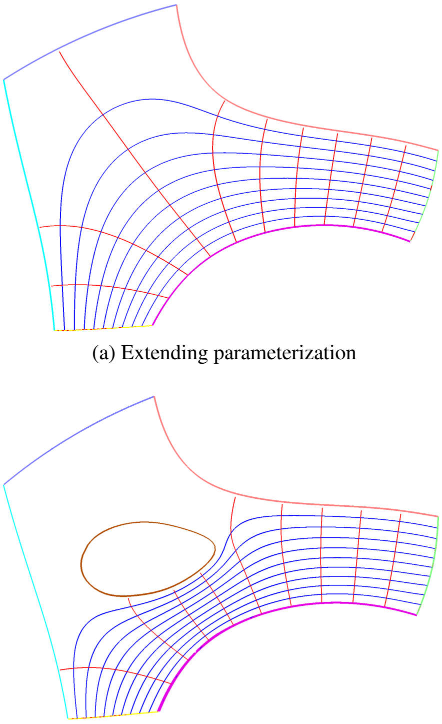
4. Algorithms to set cross-derivatives

In this section we discuss various techniques to construct cross-derivatives based on scalar weighting functions. An overview of our workflow is shown in Fig. 5.

4.1. Initial placement of ribbon control points

As it was explained earlier, we cannot determine an appropriate cross-derivative function of a multi-sided patch *locally,* using only a surface constraint from an adjacent patch. We need *global* shape information, as our cross-derivatives are supposed to adapt to the remaining (opposite) part of the patch. At first sight this seems to be unsolvable, since the surface does not exist yet, but we claim that sufficient information can be extracted from a curved domain. It is our basic assumption that there exists a domain in 2D that resembles the surface, and for each ribbon the (s, h) parameterization proportionally distributes the isolines within this domain. The examples in Fig. 6 show h-isolines of the bottom boundary distributed within an enlarged domain and in a domain with an interior hole.

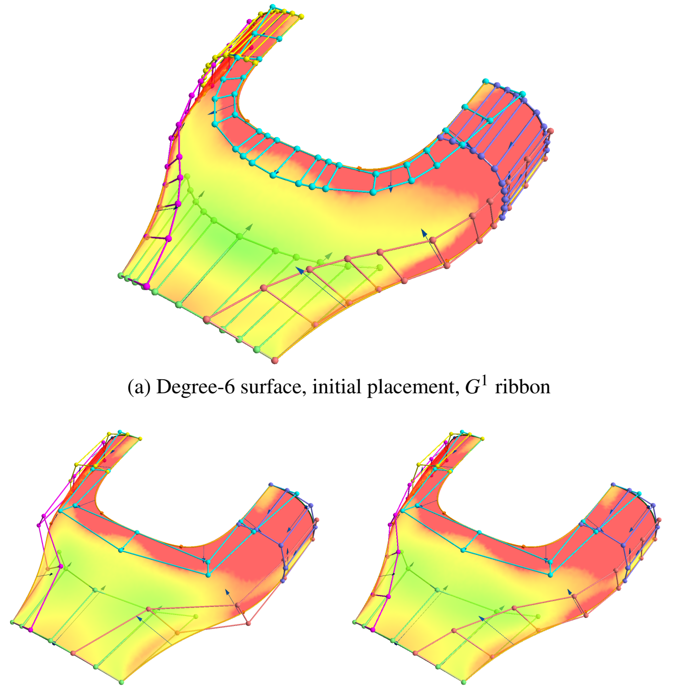
Consider a simple bicubic Bézier patch. Its $h = \frac{1}{3}$ isolines, mapped onto the surface, mimic the magnitude of the cross-derivative functions (divided by three due to the degree-3 blending functions). If the control points in the second row



(a) Extending parameterization

(b) Shrinking parameterization

Figure 6: Constant (s, h) parameter lines.



(a) Degree-6 surface, initial placement, G^1 ribbon

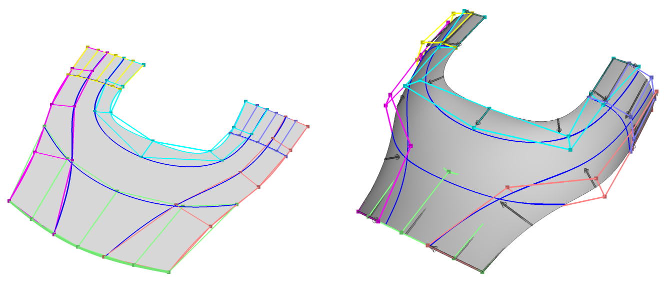
(b) Degree-3 surface, initial placement, NG^1 ribbon

(c) Degree-3 surface, initial placement, refined NG^1 ribbon

Figure 8: Initial placement with exact and approximate G^1 continuity.

are modified, the $h = \frac{1}{3}$ isolines also shift in a strongly correlated manner. This leads to our multi-sided concept. Assuming we have cubic blending functions in the cross-direction, the $h = \frac{1}{3}$ isolines of the curved domain will deliver good estimates for the magnitude of the cross-derivative functions in 3D. In other words, if we lift the related difference vectors from 2D to 3D, these produce a reasonable initial scaling for the cross-derivative functions. This is illustrated in Figure 7 for a 6-sided GBS patch.

We propose to define the scaling function $\alpha(s)$ by a simple approximative approach. Sample the boundary curve $h = 0$ and the isoline $h = \frac{1}{3}$ in the 2D domain, compute the distance l_i between associated points at the sampled parameter values s_i, and set the cross-directional scaling to $\alpha_i = l_i/\|\mathbf{D}(s_i)\|$, leading to a sequence of α_i values. Approximating these by a scalar Bézier function of a given degree or a B-spline with a given knot vector produces an appropriate scalar function $\alpha(s)$ that guarantees that the resulting cross-derivative function retains G^1 continuity. In the unlikely case when the approximation error exceeds the (typically loose) angular tolerance, the representation of $\alpha(s)$ may need to be refined.



(a) Curved domain: $h = \frac{1}{3}$ isolines, with control vectors at the Greville points of the domain boundaries

(b) Surface: $h = \frac{1}{3}$ isolines mapped to 3D, with control vectors at the boundary points, and approximate control points in the second rows

Figure 7: Isolines on a curved domain and the corresponding surface.

There may be applications where it is sufficient to keep the cross-derivative function in the form defined by Eq. (2), and the ribbons can be evaluated in a procedural manner. However, when we deal with GB or GBS patches (see Appendix A), the ribbons need to be represented by Bézier or B-spline control grids. For Bézier ribbons the use of scalar weighting functions leads to degree elevation; for B-spline ribbons, the product of the B-spline functions will lead to degree-6 ribbons over the original knots (with multiplicity). A simple and efficient method for B-spline multiplication can be found in Che et al. [34].

In many CAGD applications algebraic G^1 continuity is a crucial issue. At the same time, there are other cases where numerical continuity (NG^1) is sufficient, i.e., we only require that the angular deviation of the normal vectors along the boundary to be less than a certain tolerance. We can produce an *approximate* cross-derivative function from the exact solution, where the cubic degree and the number of control points is retained. If the tolerance criterion is not met, the knot vector can be refined.

As an illustration, Figure 8a shows exact degree-6 ribbons and their control points; Fig. 8b shows approximate degree-3 ribbons, and Fig. 8c shows a variation where a few knots are inserted to meet the prescribed angular tolerance (0.1 degree). Related data are supplemented in the next section (Table 1).

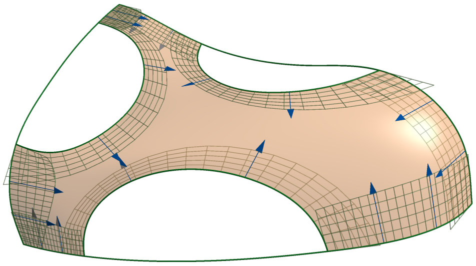


Figure 9: Control vectors.

We would like to emphasize that in our view it is hardly possible to define a single best setting of the cross derivatives, as it may depend on user requests and the type of application. However, our algorithm is capable of producing good initial solutions in the majority of practical cases, and represents a suitable basis for creating shape variations or fine-tuning surface qualities where needed. A few examples will be shown later in Section 5.

4.2. Constrained editing of ribbons

Editing ribbons may be necessary for various reasons, such as (i) improving the initial cross-derivatives, (ii) fine-tuning the shape interior, or (iii) adjusting surface qualities (isophotes, curvature distribution, etc.). As indicated earlier, constrained editing of the ribbon control points is an almost impossible task due to the fairly complex system of algebraic equations that determines G^1 continuity. Modification through $\alpha(s)$, $\beta(s)$ functions simplifies the problem, since adjusting the related coefficients, as defined by Eq. (2), provides an automatic solution to maintain continuity and modify only the involved control points. Modifying the graphs of these scalar functions would be cumbersome and artificial; in contrast, the control vector technique makes editing intuitive in 3D space, as it assigns geometric meaning to the related functions being modified in the background. As an illustration, we have assigned two internal control vectors to each boundary of the test surface in Figure 9.

While it is possible to set up a couple of control vectors and construct scalar functions that simultaneously interpolate given vectors, we prefer another step by step approach, that was found useful in our design exercises. We can place a control vector to a given point of the boundary and tweak it as required. This defines an interpolating constraint for $\alpha(s)$ and $\beta(s)$, and as they are given in B-spline form, the functions can be modified in a fairly stable way using the algorithm described by Piegl and Tiller [35, Section 11.2]. The scalar functions $\alpha(s)$ and $\beta(s)$ are modified locally by proportionally relocating two (or more) adjacent control coefficients, whose Greville abscissae surround the selected parameter value. An example is shown in Figure 11. If the selected parameter value happens to be a Greville abscissa, then only a single control point of $\alpha(s)$ and $\beta(s)$ will be modified. A similar algorithm can be applied for the local modifications of $\alpha(s)$ and $\beta(s)$ given in Bézier form, as well.

In Figure 10 we present how editing by control vectors proceeds. We also visualized the cross-derivative surfaces multiplied by $\frac{1}{3}$, superimposed onto the multi-sided surface. We modified the bottom side; observe how a single control vector pulls the surrounding exact (degree-6) and approximate (degree-3) control points.

5. Discussion

In this section we evaluate the proposed constrained editing technique through various test examples.

5.1. Bypassing common problems

The first sequence illustrates how multi-sided surfacing can go wrong for a setback vertex blend surface with eight boundary curves. Figure 12a shows that when a convex polygonal domain is used, the patch deteriorates. In Figure 12b we use a nicely parameterized curved domain, but the ribbon magnitudes are excessive, leading to wrinkles in the interior. In Figure 12c, the proposed initial setting of cross-derivatives produces a nice patch.

5.2. Initial cross-derivatives

The six-sided patch in Figure 13 represents three variants: (a) narrow ribbons (flat part in the middle), (b) large ribbons (wrinkling), and (d) automatically set ribbons by the initial placement algorithm. In (c) the curved domain is shown, observe the $h = \frac{1}{3}$ isolines and the coloring that indicates where the individual ribbons dominate.

Initial cross-derivative settings are shown for two simple test objects with isophote lines (Fig. 14).

5.3. Editing

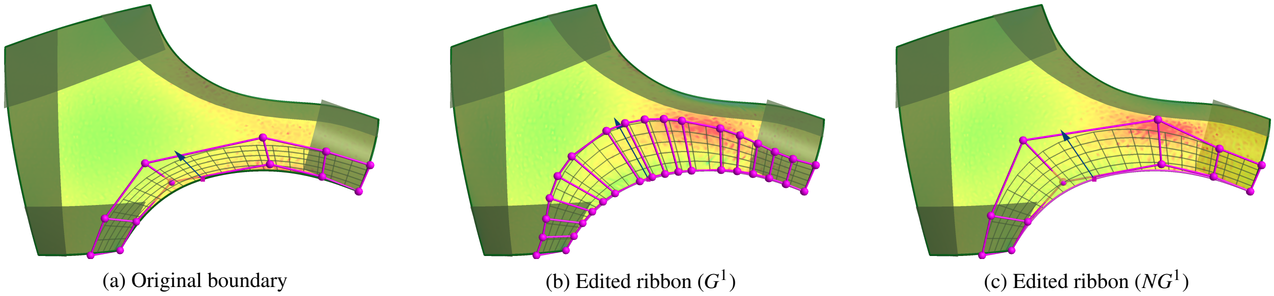
The six-sided vertex blend in Fig. 15 is highly curved, and the global Gaussian curvature is high, and consequently the flattened domain is relatively strongly distorted. In these cases the initial cross-derivative setting will generally produce weak ribbons and dragging them towards the middle is often necessary. The curvature in Figure 15a shows a flat interior, while after editing the expected distribution is obtained (Figure 15b).

In Figure 16 we edit the surface seen on Fig. 10 by adding a hole loop that lies on an extruded surface. Using a few control vectors the transition between the middle part and the outside boundaries is modified.

5.4. Approximate ribbons (NG^1)

Returning to Figures 8b–8c, Table 1 shows the results of numerical G^1 approximation. We have set the angular tolerance to 0.1 degree, and according to the data, Ribbons 3 and 5 (on the left and right sides of the patch) are not accurate enough. For this experiment, we have doubled the knots, and the accuracy improved to a great extent. The approximate ribbons are computed from the exact cross-derivatives by regularized least-squares fitting. It depends on the context whether to prefer these to the exact ribbons (less control points matching degree-3 ribbons or more control points with degree-6 and algebraic G^1 continuity).

Another numerical example is given by analyzing the hole loop in Fig. 16. The approximate representation had an angular accuracy of 0.39 degrees, which was reduced after inserting knots to 0.041. In this particular case, the algebraic degree-6 ribbon has 2×35 control points, the approximate ribbon 2×14.


(a) Original boundary

(b) Edited ribbon (G^1)

(c) Edited ribbon (NG^1)

Figure 10: Editing a ribbon with a control vector.

Ribbon	Mean deviation	Max deviation
R_1	0.0007	0.0029
R_2	0.0004	0.0010
R_3	0.1044	0.1946
R_3^*	→0.0029	→0.0078
R_4	0.0086	0.0199
R_5	0.1075	0.1956
R_5^*	→0.0314	→0.0081
R_6	0.0001	0.0003

Table 1: G^1 approximation errors showing angular deviations (in degrees). Ribbons after knot refinement are shown with a star.

5.5. T-nodes

The proposed modeling paradigm, i.e., creating patches through cross-derivative constraints, naturally supports the creation of patchworks with T-nodes. Imagine that a cross-derivative function has been associated with a full curve and is cut into pieces. Then the related portions are transferred to the patches, and G^1 continuity will be ensured along the whole curve. Three T-nodes are shown on the left side of the test example in Figure 17.

5.6. Complex sheet metal parts

The complex test part in Figs. 18a–18b illustrates that in several cases one cannot associate a natural flow of isolines with the shape, and by trimming it is quite difficult to meet prescribed positional and tangential constraints. Here we have three concave boundary curves, being explicitly defined by their cross-derivatives. The patch is defined by a single parametric equation and represents a nice blend between the boundaries, though admittedly the isophotes should be enhanced at certain places.

A test part is shown in Fig. 18c. The base patch has ten boundaries, and adding two hole loops in the middle naturally changes the patch interior. This surface is also defined by a single parametric equation and the cross-derivatives were obtained by initializing and editing ribbons based on the given constraint surfaces. A more complex example is shown in Fig. 19 that defines a concept car, consisting of eight 5-sided and six 4-sided patches, also demonstrating T-node connections.

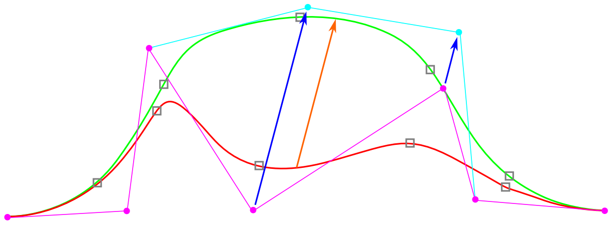


Figure 11: A control vector modifies the scalar function at a given position; the two adjacent control coefficients are relocated accordingly. The small squares show the positions of the Greville abscissae.

5.7. Limitations and future work

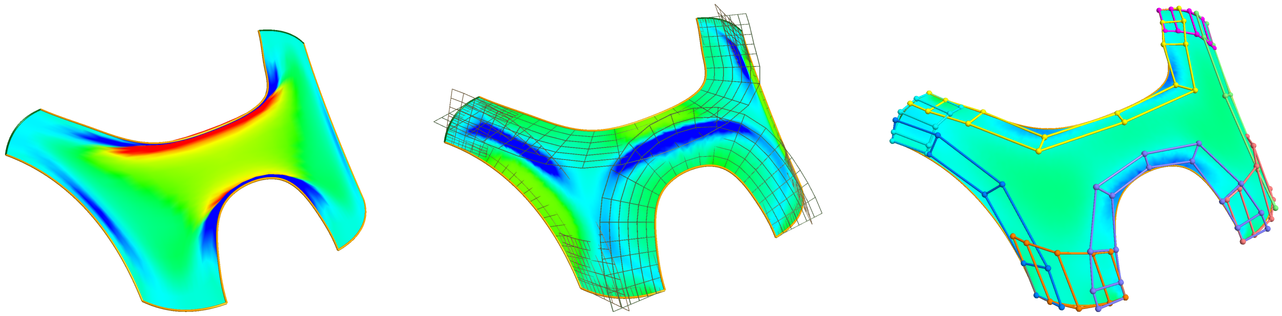
There are several open issues and deficiencies in the current scheme. The most crucial is ribbon parameterization within a curved domain, which fundamentally determines the quality of multi-sided, multi-connected patches. Harmonic functions represent one particular solution; their evaluation is computationally demanding, but multiresolution methods can speed it up considerably. While in the great majority of cases harmonic coordinates provide nice, evenly distributed parametric structures, in certain cases they behave in an unexpected manner. This requires further analysis and search for alternative parameterizations.

The initial setting of cross-derivatives produces ‘weak’ ribbons for highly curved multi-sided patches, and we search for enhanced algorithms to detect and compensate this effect.

It must also be noted, that ribbon-based patches have a limitation that in highly complex cases the information coming from the ribbons may not be sufficient to properly define the surface interior. In these cases, additional constraints or control structures need to be supplemented—this is also part of an ongoing research project.

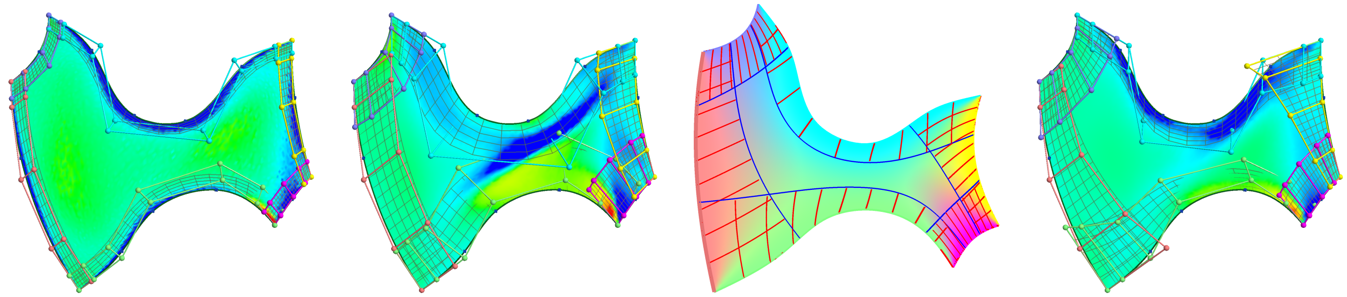
Conclusion

We have investigated constrained modeling to set cross-derivatives for multi-sided, multi-connected patches. Our work builds (i) on curved domains that represent a moderately distorted planar map of the surface to be constructed, and (ii) on scalar weighting functions (scaling and shearing) to set the cross-derivatives of the ribbons that determine the patch. The proposed initial placement method is based on the parameterization of the ribbons within a curved domain. The proposed



(a) Poor quality due to convex domain (b) Poor quality due to overshooting ribbons (c) Good quality due to proper initial settings

Figure 12: Setback vertex blend with poor and appropriate settings.



(a) Narrow ribbons (b) Large ribbons (c) Curved domain (d) Initialized ribbons

Figure 13: Effects of different ribbon magnitudes.

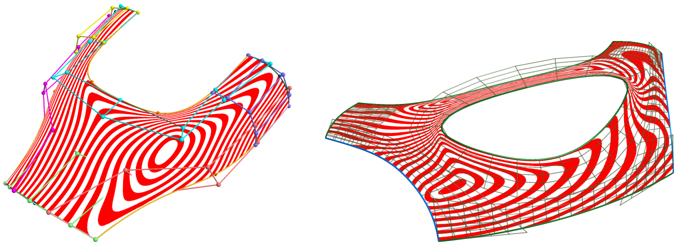
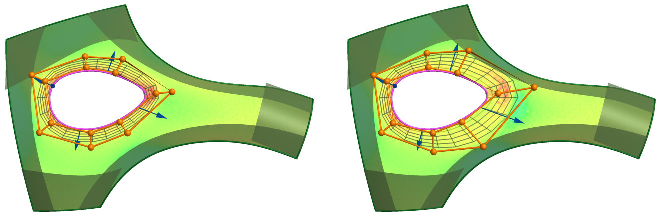
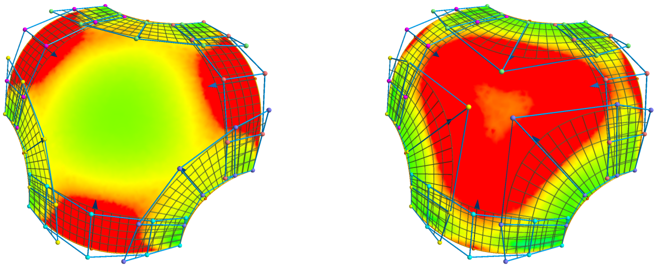


Figure 14: Test parts with isophotes.



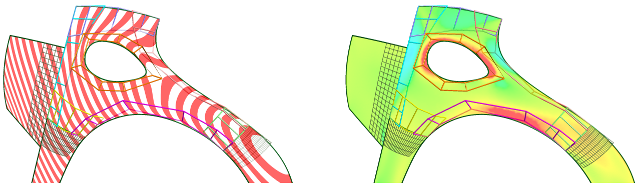
(a) Before (b) After

Figure 16: Editing a periodic ribbon.



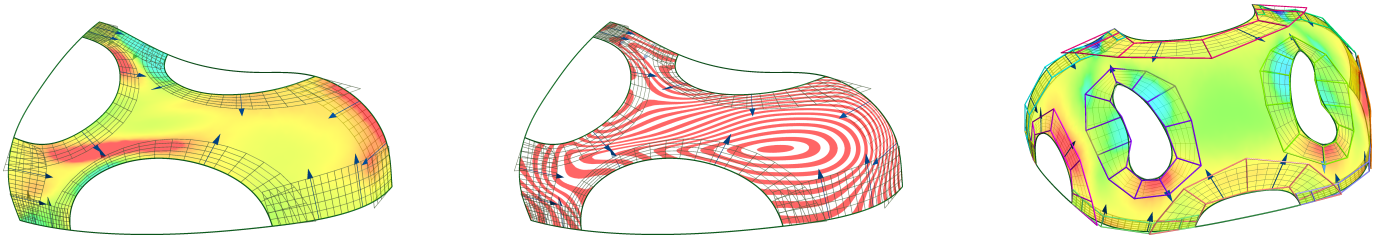
(a) Initial setting (b) Edited

Figure 15: Ribbon editing to ensure even curvature in the middle.



(a) Contouring (b) Mean curvature

Figure 17: Example with T-nodes.



(a) Part 1 with mean curvature

(b) Part 1 with isophotes

(c) Part 2 with mean curvature

Figure 18: Complex examples.

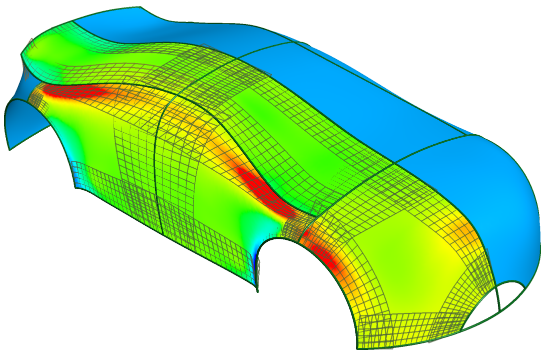


Figure 19: Concept car model.

editing operations are performed by control vectors that simultaneously relocate a collection of control points. These methods automatically guarantee G^1 continuity to the surrounding patches and are useful for constructing good quality multi-sided patches.

Acknowledgments

This project has been supported by the Hungarian Scientific Research Fund (OTKA, No. 124727: Modeling general topology free-form surfaces in 3D). The authors would like to acknowledge the significant programming contribution of György Karikó for extending our prototype surfacing system Sketches (ShapEx Ltd., Budapest).

Appendix A.

We summarize the formulation of GB and GBS patches in a nutshell; for details, see Várady et al. [1] and Vaitkus et al. [2]. These interpolating patches are defined by a collection of open or periodic ribbon surfaces that prescribe positions and cross-derivatives along the surface boundaries. The i-th ribbon $\mathbf{R}_i$ is defined by $(d_i + 1) \times (e_i + 1)$ control points that correspond to the longitudinal and cross directions, respectively. Its equation is given as

$$\mathbf{R}_i(s_i, h_i) = \sum_{j=0}^{d_i} \sum_{k=0}^{e_i} \mathbf{C}^i_{j,k} \omega^i_{j,k}(s_i, h_i). \tag{A.1}$$

Here $s_i, h_i \in [0, 1]$ denote the local ribbon parameters along and across the boundary. For GB patches the weighting function is the product of two Bernstein basis functions, for GBS patches the product of B-spline and Bernstein basis functions (the former defined over a knot vector ξ_i with degree p_i), i.e.,

$$\omega^i_{j,k}(s_i, h_i) = B^{d_i}_j(s_i) B^{e_i}_k(h_i) \quad \text{and} \tag{A.2}$$

$$\omega^i_{j,k}(s_i, h_i) = N^{\xi_i, p_i}_j(s_i) B^{e_i}_k(h_i), \tag{A.3}$$

respectively.

These patches are defined over a *curved domain* in the (u, v) plane. For each ribbon there is a longitudinal and a distance parameter (s_i, h_i) that determine how the points of the ribbon are mapped onto the multi-sided patch.

The multi-sided patch equation (Eq. A.7) sums *weighted ribbons* $\mathbf{R}^*_i$, which retain the control points of the ribbons $\mathbf{R}_i$, but employ modified blending functions:

$$\mathbf{R}^*_i(s_i, h_i) = \sum_{j=0}^{d_i} \sum_{k=0}^{e_i} \mathbf{C}^i_{j,k} \Omega^i_{j,k}(s_i, h_i), \tag{A.4}$$

where

$$\Omega^i_{j,k}(s_i, h_i) = \mu^i_j(h_i) B^{d_i}_j(s_i) B^{2e_i+1}_k(h_i) \quad \text{and} \tag{A.5}$$

$$\Omega^i_{j,k}(s_i, h_i) = \mu^i_j(h_i) N^{\xi_i, p_i}_j(s_i) B^{2e_i+1}_k(h_i), \tag{A.6}$$

respectively. For both GB and GBS, Bernstein functions of degree $2e_i + 1$ are applied in the cross direction. These ensure positional and cross-derivative interpolation with G^{e_i} continuity on side i, and guarantee that the weighted ribbons $\mathbf{R}^*_i$ disappear on the distant sides of the domain. Thus for G^1 we use cubic, for G^2 quintic basis functions.

Each Ω^i contains a *rational correction term* $\mu^i_j(h_i)$, similar to those defined by Gregory [36], that ensures the reproduction of the given ribbon on the i-th side $(h_i = 0)$, and forces $\mathbf{R}^*_i$ to vanish on the neighboring sides $i - 1$ and $i + 1$. The multi-sided patch $\mathbf{S}(u, v)$ is defined as the sum of n weighted ribbons. In order to ensure the convex combination property, we normalize the patch by the sum of the corrected blending functions*:*

$$\mathbf{S}(u, v) = \frac{1}{\Omega_\Sigma(u, v)} \cdot \sum_{i=1}^{n} \mathbf{R}^*_i(s_i, h_i), \tag{A.7}$$

$$\Omega_\Sigma(u, v) = \sum_{i=1}^{n} \sum_{j=0}^{d_i} \sum_{k=0}^{e_i} \Omega^i_{j,k}(s_i, h_i). \tag{A.8}$$

The curved domains of the GB and GBS patches are generated with the method detailed in Várady et al. [1], developing the boundary curves into the tangent planes of the interpolants. For the parameterization we apply *harmonic functions*; further details can be found in Vaitkus et al. [2].

Note that this representation is not directly compatible with current CAD standards, but a conversion to NURBS can be performed by approximation.

References

[1] Várady, T, Salvi, P, Vaitkus, M, Sipos, Á. Multi-sided Bézier surfaces over curved, multi-connected domains. Computer Aided Geometric Design 2020;78:101828. doi:10.1016/j.cagd.2020.101828.

[2] Vaitkus, M, Várady, T, Salvi, P, Sipos, Á. Multi-sided B-spline surfaces over curved, multi-connected domains. Computer Aided Geometric Design 2021;89:102019. doi:10.1016/j.cagd.2021.102019.

[3] Salvi, P, Várady, T, Rockwood, A. Ribbon-based transfinite surfaces. Computer Aided Geometric Design 2014;31(9):613–630. doi:10.1016/j.cagd.2014.06.006.

[4] Karčiauskas, K, Peters, J. Can bi-cubic surfaces be class A? Computer Graphics Forum 2015;34(5):229–238. doi:10.1111/cgf.12711.

[5] Malraison, P. *N*-sided surfaces: A survey. In: Curve and Surface Design. Vanderbilt University Press; 2000, p. 247–256.

[6] Várady, T, Rockwood, A, Salvi, P. Transfinite surface interpolation over irregular *n*-sided domains. Computer-Aided Design 2011;43(11):1330–1340. doi:10.1016/j.cad.2011.08.028.

[7] Martin, F, Reif, U. Trimmed spline surfaces with accurate boundary control. In: Geometric Challenges in Isogeometric Analysis. Springer; 2022, p. 123–148. doi:10.1007/978-3-030-92313-6_6.

[8] Sabin, MA, Fellows, C, Kosinka, J. CAD model details via curved knot lines and truncated powers. Computer-Aided Design 2022;143:103137. doi:10.1016/j.cad.2021.103137.

[9] Kato, K. Generation of *n*-sided surface patches with holes. Computer-Aided Design 1991;23(10):676–683. doi:10.1016/0010-4485(91)90020-w.

[10] Sabin, M. Further transfinite surface developments. In: The Mathematics of Surfaces; vol. 8. Information Geometers; 1998, p. 161–174.

[11] Peters, J. Geometric continuity. In: Handbook of Computer Aided Geometric Design. Elsevier; 2002, p. 193–227. doi:10.1016/B978-0-444-51104-1.X5000-X.

[12] Kiciak, P. Geometric Continuity of Curves and Surfaces. Synthesis Lectures on Visual Computing; Morgan & Claypool Publishers; 2016.

[13] Chiyokura, H, Kimura, F. Design of solids with free-form surfaces. In: Proceedings of the 10th annual conference on Computer Graphics and Interactive Techniques. 1983, p. 289–298. doi:10.1145/800059.801160.

[14] Hermann, T, Lukács, G. A new insight into the G^n continuity of polynomial surfaces. Computer Aided Geometric Design 1996;13(8):697–707. doi:10.1016/0167-8396(96)00005-2.

[15] Hoschek, J, Lasser, D. Fundamentals of Computer Aided Geometric Design. AK Peters, Ltd.; 1993.

[16] Renner, G. Polynomial *n*-sided patches. In: Curves and Surfaces. Academic Press; 1991, p. 407–410. doi:10.1016/B978-0-12-438660-0.50064-9.

[17] Shi, KL, Zhang, S, Zhang, H, Yong, JH, Sun, JG, Paul, JC. G^2 B-spline interpolation to a closed mesh. Computer-Aided Design 2011;43(2):145–160. doi:10.1016/j.cad.2010.10.004.

[18] Che, X, Liang, X, Li, Q. G^1 continuity conditions of adjacent NURBS surfaces. Computer Aided Geometric Design 2005;22(4):285–298. doi:10.1016/j.cagd.2005.01.001.

[19] Peters, J, Fan, J. On the complexity of smooth spline surfaces from quad meshes. Computer Aided Geometric Design 2010;27(1):96–105. doi:10.1016/j.cagd.2009.09.003.

[20] Shi, X, Wang, T, Yu, P. A practical construction of G^1 smooth biquintic B-spline surfaces over arbitrary topology. Computer-Aided Design 2004;36(5):413–424. doi:10.1016/S0010-4485(03)00111-8.

[21] Mosbach, D, Schladitz, K, Hamann, B, Hagen, H. A local approach for computing smooth B-spline surfaces for arbitrary quadrilateral base meshes. Journal of Computing and Information Science in Engineering 2022;22(1). doi:10.1115/1.4051121.

[22] Yuksel, C, Schaefer, S, Keyser, J. Parameterization and applications of Catmull–Rom curves. Computer-Aided Design 2011;43(7):747–755. doi:10.1016/j.cad.2010.08.008.

[23] Antonelli, M, Beccari, CV, Casciola, G. High quality local interpolation by composite parametric surfaces. Computer Aided Geometric Design 2016;46:103–124. doi:10.1016/j.cagd.2016.06.005.

[24] Shirman, LA, Séquin, CH. Procedural interpolation with geometrically continuous cubic splines. Computer-Aided Design 1992;24(5):267–277. doi:10.1016/0010-4485(92)90080-T.

[25] Peters, J. Splines for meshes with irregularities. The SMAI Journal of Computational Mathematics 2019;5:161–183. doi:10.5802/smai-jcm.57.

[26] Hughes, TJ, Sangalli, G, Takacs, T, Toshniwal, D. Smooth multi-patch discretizations in isogeometric analysis. In: Handbook of Numerical Analysis; vol. 22. Elsevier; 2021, p. 467–543. doi:10.1016/bs.hna.2020.09.002.

[27] Chiyokura, H. Localized surface interpolation method for irregular meshes. In: Advanced Computer Graphics: Proceedings of Computer Graphics Tokyo'86. 1986, p. 3–19. doi:10.1007/978-4-431-68036-9_1.

[28] Konno, K, Takamura, T, Chiyokura, H. A new control method for free-form surfaces with tangent continuity and its applications. In: Scientific Visualization of Physical Phenomena. Springer; 1991, p. 435–456. doi:10.1007/978-4-431-68159-5_25.

[29] Shen, J, Kosinka, J, Sabin, M, Dodgson, N. Converting a CAD model into a non-uniform subdivision surface. Computer Aided Geometric Design 2016;48:17–35. doi:10.1016/j.cagd.2016.07.003.

[30] Shirman, LA, Séquin, CH. Local surface interpolation with shape parameters between adjoining Gregory patches. Computer Aided Geometric Design 1990;7(5):375–388. doi:10.1016/0167-8396(90)90001-8.

[31] Konno, K, Chiyokura, H. An approach of designing and controlling free-form surfaces by using NURBS boundary Gregory patches. Computer Aided Geometric Design 1996;13(9):825–849. doi:10.1016/S0167-8396(96)00012-X.

[32] Konno, K, Chiyokura, H. G^1 and G^2 surface interpolation over curve meshes and its shape control. International Journal of Shape Modeling 1996;2(01):1–20. doi:10.1142/S0218654396000026.

[33] Zou, M, Ju, T, Carr, N. An algorithm for triangulating multiple 3D polygons. Computer Graphics Forum 2013;32(5):157–166. doi:10.1111/cgf.12182.

[34] Che, XJ, Farin, G, Gao, ZH, Hansford, D. The product of two B-spline functions. In: Advanced Materials Research; vol. 186. 2011, p. 445–448. doi:10.4028/www.scientific.net/AMR.186.445.

[35] Piegl, L, Tiller, W. The NURBS Book. Springer Science & Business Media; 1996. doi:10.1007/978-3-642-97385-7.

[36] Gregory, JA. Smooth interpolation without twist constraints. In: Computer Aided Geometric Design. Academic Press; 1974, p. 71–87. doi:10.1016/B978-0-12-079050-0.50009-6.