Smooth Bi-3 Spline Surfaces with fewest knots

Jianhua Fan, Jörg Peters

Department of CISE, University of Florida, USA 32603

Abstract

Converting a quadrilateral input mesh into a C^1 surface with one bi-3 tensor-product spline patch per facet is a classical challenge. We give explicit local averaging formulas for the spline control points. Where the quadrilateral mesh is not regular, the patches have two internal double knots, the least number and multiplicity to always allow for an unbiased G^1 construction.

Key words: spline surfaces, bicubic, construction, geometry continuity, Catmull-Rom splines.

1. Introduction

While smooth surfaces of degree bi-3 can be generated from quadrilateral meshes by recursively applying subdivision [CC78], using a finite number of polynomial pieces is often preferred due to existing CAD standards or to support parallelism when rendering.

When associating one bi-cubic tensor-product spline patch with each facet of a quadrilateral mesh, the main challenge comes from converting *extraordinary* quads, i.e. quads that have one or more vertices of valence $n \neq 4$ (*red* quads in Figure 1 (d)). Regular quads have only vertices of valence 4 and therefore allow interpretation of the surrounding 4×4 grid of vertices as control points of bi-cubic B-spline surface (*gold* quads in Figure 1 (d)).

For localized smooth surface constructions that only access a fixed, small neighborhood of the quad, we established in [PF09] the minimal number and multiplicity of knots that allows for one bi-3 spline patch per quad. This lower bound is exactly met by our construction in this paper. Here we present explicit *B-spline (NURBS) control point* formulas for
$--$ a construction with one degree bi-3 tensor-product spline patch per quad;
$--$ at most two interior double knots, and
$--$ a complete smoothness analysis.
Each tensor-product spline patch has, per tensor-direction, either no interior knot (and represents a piece of a standard C^2 spline complex) or two interior double knots. Vertex position and normal are either derived from the mesh or may be explicitly prescribed.

Section 2 gives explicitly input, output, construction stages and formulas with illustrations of the B-spline coefficients. Section 3 formally verifies that

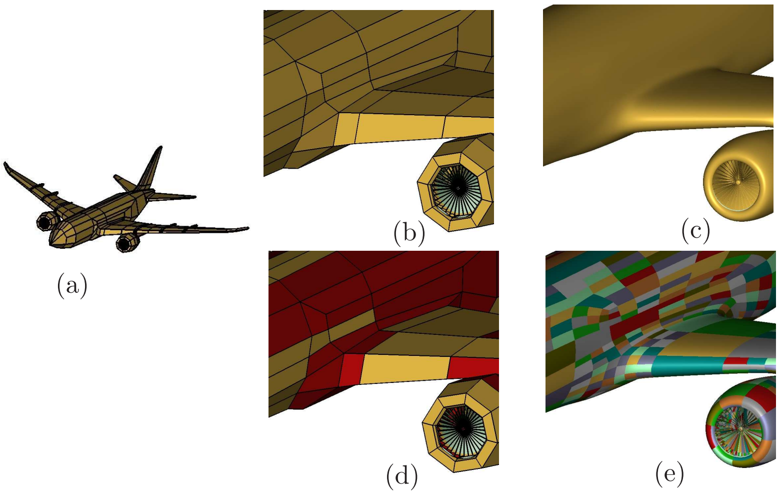


Figure 1: Bi-3 **tensor-product spline construction.** (a) Quad mesh with (b) enlarged section and (c) surface generated by the algorithm. (d) *Red* quads are extraordinary with possibly several non-4-valent vertices. Regular *gold* quads correspond to bi-3 polynomial patches that join C^2 with their neighbors, also across T-junctions. (e) A minimal number of bi-3 polynomial pieces (random coloring).

the construction satisfies the G^1 continuity constraints, i.e., for generic input, results in a surface that is at least C^1 surface everywhere. Moreover transitions across edges connecting 4-valent vertices (regular edges) are C^2. Section 4 examines the quality of the resulting surfaces by examples and Section 5 points to generalizations such as using more than the minimal number of knots in the spline patches.

1.1. Literature on bi-3 C^1 surface constructions

Generalizing bicubic (bi-3) tensor-product spline patches to C^1 surfaces on general quad meshes is a classic challenge with a more than 30 year history [Gre74, Bez77, vW86, Pet91, GZ94]. More recently, a number of publications have re-addressed this challenge in the context of the impending ability of GPUs to tessellate and adaptively evaluate patched polynomial surfaces at animation speeds. Loop and Schaefer [LS08] propose bi-cubic C^0 surfaces with surrogate tangent patches to convey the impression of smoothness via lighting. Myles *et al.* [MYP08] perturbed a bi-cubic base patch near non-4-valent vertices by coefficients of a (5,5) patch to obtain a C^1 surface. PCCM [Pet00] generates smooth bi-cubic surfaces. PCCM requires up to two steps of Catmull-Clark subdivision to separate non-4-valent vertices resulting in at least a 4×4 arrangement of polynomial patches per quad. [Pet01] pointed out that PCCM in its original form can have shape problems at higher-order (e.g. monkey) saddles. The 3×3 arrangement of Bézier patches per quadrilateral in our conference paper [FP08] has a re-parameterization different from PCCM that avoids the PCCM shape problem. The construction we present below is very similar to [FP08] (after conversion to B-spline form) but subtly different. It guarantees C^2 continuity across all *regular edges* (where both end vertices have valence 4), while [FP08] guarantees only C^1 continuity (see Figure 5(c) *vs* (d)).

2. The spline construction for extraordinary quads

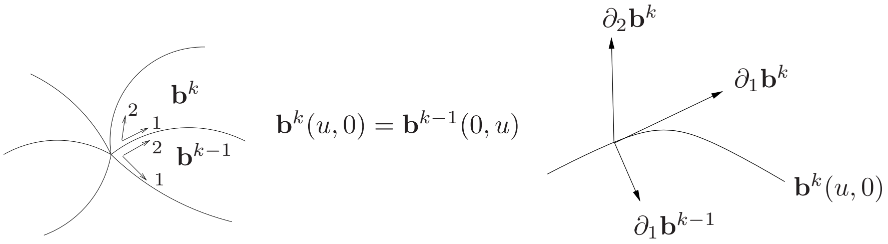


Figure 2: G^1 **constraints:** (*left*) n patches joining. (*right*) Coplanar partial derivatives along the kth boundary curve $\mathbf{b}^k(u,0)$.

Consider n patches that are internally (parametrically) C^1

$$\mathbf{b}^k : [0..1]^2 \subsetneq \mathbb{R}^2 \to \mathbb{R}^3, \qquad k = 1, \dots, n, \tag{1}$$

and that meet at a central point $\mathbf{b}^k(0,0)$ and such that $\mathbf{b}^k(u,0) = \mathbf{b}^{k-1}(0,u)$ as displayed in Figure 2. For every pair of patches sharing a curve, we enforce the *unbiased* G^1 *constraints*

$$\partial_2\mathbf{b}^k(u,0) + \partial_1\mathbf{b}^{k-1}(0,u) = \alpha^k(u)\partial_1\mathbf{b}^k(u,0), \tag{2}$$

where ∂_ℓ means differentiation with respect to the ℓth argument and $\alpha^k : [0..1] \to \mathbb{R}$ is a univariate linear or quadratic *scalar function.* That is, the constraints are logically symmetric in that exchanging the index $k \leftrightarrow k^+ := (k+1) \bmod n^0$ does not change the constraints. If $\alpha^k = 0$, the constraints enforce *parametric* C^1 *continuity.* When the quad is covered by a spline with two internal knots, i.e. a three by three arrangement of C^1-connected polynomial pieces, we use a subscript ℓ to refer to the ℓth piece α^k_ℓ, $\ell = 0, 1, 2$ of α^k along a boundary.

To delineate each bi-3 spline patch, we place 4-fold knots at the ends of the knot sequence. To cope with extraordinary quads with one or more vertices of valence $n \neq 4$, [PF09, Thm 1]proves that at least two double internal knots are needed. Scaling to multiples of 3 yields a knot sequence $\kappa := (\kappa_0, \kappa_1, \dots \kappa_{11}) = (0,0,0,0,3,3,6,6,9,9,9,9)$ that can nicely be used to associate control points with their Greville abscissae γ (cf. Figure 3 *right*). That is for the kth patch surrounding the central point,

$$\mathbf{b}^k(u,v) := \sum_{i=0}^{7}\sum_{j=0}^{7} \mathbf{b}^k_{\gamma_i,\gamma_j} N_i(u) N_j(v), \quad \gamma := (\gamma_0, \dots \gamma_7) = (0,1,2,4,5,7,8,9) \tag{3}$$

where N_i is the degree 3 B-spline with knot sequence $\kappa_i, \dots \kappa_{i+4}$. Below, we will also refer to the bi-3 (= bi-cubic) Bernstein-Bézier form (short: *BB-form*) of a polynomial

$$\sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{c}_{\gamma_i,\gamma_j} f_i(u) f_j(v), \quad f_\ell(t) := \binom{3}{\ell}(1-t)^{3-\ell}t^\ell.$$

To convert a uniform spline with B-spline coefficients $\mathbf{p}_{ij}$ to the BB-form we need only specify

$$\begin{aligned} 36\mathbf{c}_{00} &= 16\mathbf{p}_{11} + 4(\mathbf{p}_{21} + \mathbf{p}_{12} + \mathbf{p}_{01} + \mathbf{p}_{10}) + (\mathbf{p}_{22} + \mathbf{p}_{02} + \mathbf{p}_{00} + \mathbf{p}_{20}), \\ 18\mathbf{c}_{10} &= 8\mathbf{p}_{11} + 2(\mathbf{p}_{10} + \mathbf{p}_{12}) + 4\mathbf{p}_{21} + \mathbf{p}_{22} + \mathbf{p}_{20}, \\ 9\mathbf{c}_{11} &= 4\mathbf{p}_{11} + 2(\mathbf{p}_{21} + \mathbf{p}_{12}) + \mathbf{p}_{22} \end{aligned} \tag{4}$$

since the remaining 13 formulas are obtained by combinatorial symmetry.

2.1. The Algorithm

We factor the algorithm into five localized stages that are explained in detail in Sections 2.4–2.8. The coefficients of a stage can be determined in parallel, for all quads, respectively all vertices.

Algorithm
For each quad, we

1. compute a bi-3 tensor product patch $\mathbf{q}$ in Bernstein-Bézier form;
   If the quad is regular, set $\mathbf{b} = \mathbf{q}$ and STOP,
   else insert, equally spaced, two repeated knots in the u and the v direction.
2. adjust tangents to enforce (2);
3. complete the boundary to enforce (2);
4. modify the first interior layer of control points to enforce (2);
5. compute the remaining interior control points by averaging.

2.2. Control points

Below we give the formulas for the B-spline control points $\mathbf{b}^k_{\gamma_i\gamma_j}$. The presentation is intentionally terse to facilitate implementation. We do however mention the choice of α^k to hint at the derivation and the proof of smoothness that follow in Section 3.

In the figures illustrating the formulas, older, already computed control points are presented as gray disks; the new control points, computed by the formulas, are shown as black disks. Only formulas for a few control points need to be shown since the remaining ones are obtained **symmetrically**. By this we mean the (obvious) *combinatorial symmetries* where formulas are obtained by exchanging the subscripts (diagonal flip of a quad $\mathbf{b}^k_{ij} \leftrightarrow \mathbf{b}^k_{ji}$) or increasing the superscript (rotation of quad $\mathbf{b}^k_{ij} \leftrightarrow \mathbf{b}^{k^+}_{ij}$ where $k^+ := k+1 \mod n^0$) or switching start and end-point along an edge (flip across the midpoint: $\mathbf{b}^k_{i0} \leftrightarrow \mathbf{b}^k_{9-i,0}$).

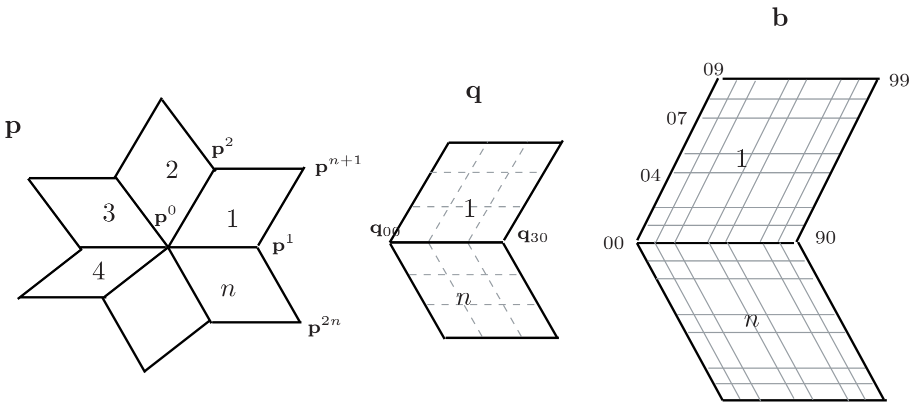


Figure 3: **Layout and Indexing.** (*left*) Local input mesh $\mathbf{p}$. (*middle*) $\mathbf{q}$ patch initialization.(*right*) NURBS control point indices of a patch $\mathbf{b}$.

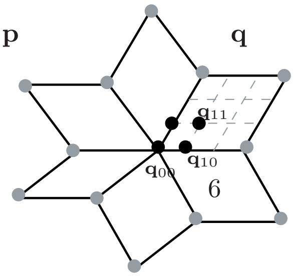
2.3. Input and Output

[Input: p]
(Figure 3 *left*) A vertex $\mathbf{p}^0$ of valence $n := n^0$ surrounded by vertices $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^{2n}$ forming n quads. The valence of the direct neighbors $\mathbf{p}^k \in \{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n\}$ of $\mathbf{p}^0$ is denoted n^k.

[Output: b]
(Figure 3 *middle*, *right*) If the quad is ordinary, the output is a bi-3 patch $\mathbf{q}$ (Figure 3 *middle*) in BB-form, i.e. a tensor-product spline patch with knots $\kappa := (0, 0, 0, 0, 9, 9, 9, 9)$ in each direction.
Otherwise, the output spline patch $\mathbf{b}$ (Figure 3 *right*) has the knot sequence $\kappa := (0, 0, 0, 0, 3, 3, 6, 6, 9, 9, 9, 9)$. Section 5 discusses what to do when more than this minimal number of knots are wanted.
Each tensor-product spline patch of degree bi-3 per quad joines its neighbors to form a smooth surface with adjacent patches satisfying unbiased G^1 constraints. Patches join C^2 across edges where both end-points have valence 4, also across *T-junctions* (where one spline corresponding to a regular quad has no knot but the other does since it corresponds to an irregular quad).

2.4. **Step 1** *Bézier Patch*

We use formula (4) to derive all points, except that, by default, we generalize the formula of $\mathbf{q}^k_{00}$ to the well-known limit point of Catmull-Clark subdivision:



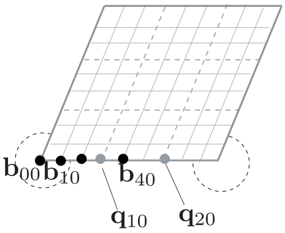
$$\mathbf{q}_{00}^k := \frac{\sum_{l=1}^n \left(n\mathbf{p}^0 + 4\mathbf{p}^l + \mathbf{p}^{n+l}\right)}{n(n+5)}, \tag{5}$$

$$\mathbf{q}_{10}^k := \frac{8\mathbf{p}^0 + 4\mathbf{p}^k + 2\mathbf{p}^{k+1} + 2\mathbf{p}^{k-1} + \mathbf{p}^{n+k} + \mathbf{p}^{n+k-1}}{18} \tag{6}$$

$$\mathbf{q}_{11}^k := \frac{1}{9}(4\mathbf{p}^0 + 2(\mathbf{p}^k + \mathbf{p}^{k+1}) + \mathbf{p}^{n+k}). \tag{7}$$

Symmetric construction of the other three corners of the quad yields coefficients $\mathbf{q}_{ij}^k$ that we interpret as the control points of a Nurbs patch with knots sequence (0,0,0,0,9,9,9,9). If the quad is regular, then we stop and output the patch $\mathbf{b} := \mathbf{q}$. We note that if we apply the remaining steps of the algorithm, we can choose $\mathbf{q}_{00}^k$ to be some other point that the surface should interpolate, e.g. the original mesh point $\mathbf{p}^0$ and some other tangents. This is illustrated in Figure 6 (d).

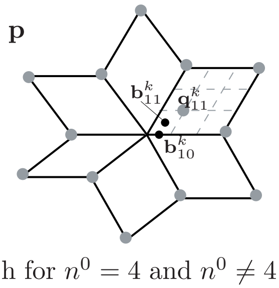
If the quad is not regular, insert two double knots (first in the u then in the v direction) to obtain



$$\begin{aligned}
\mathbf{b}_{00}^k &:= \mathbf{q}_{00}^k, \\
\mathbf{b}_{20}^k &:= \frac{4}{9}\mathbf{q}_{00}^k + \frac{4}{9}\mathbf{q}_{10}^k + \frac{1}{9}\mathbf{q}_{20}^k, \\
\mathbf{b}_{40}^k &:= \frac{4}{27}\mathbf{q}_{00}^k + \frac{12}{27}\mathbf{q}_{10}^k + \frac{9}{27}\mathbf{q}_{20}^k + \frac{2}{27}\mathbf{q}_{30}^k.
\end{aligned}$$

2.5. **Step 2.** *Tangents*

[if $n^0 = 4$] $\mathbf{b}_{10}^k := \frac{2}{3}\mathbf{q}_{00}^k + \frac{1}{3}\mathbf{q}_{10}^k$, otherwise set $\mathbf{b}_{10}^k$ as follows to lie in the tangent plane spanned by $\mathbf{e}_1$ and $\mathbf{e}_2$:



$$\mathbf{b}_{10}^k := \mathbf{b}_{00}^k + \frac{\mathbf{e}_1 c_n^k + \mathbf{e}_2 s_n^k}{3}, \mathbf{e}_i := \frac{\sigma_n}{3(2+\omega_n)} \sum_{l=1}^n \left(\delta_i \mathbf{p}^l + \beta_i \mathbf{p}^{n+l}\right)$$

$$c_n^k := \cos\frac{2\pi k}{n}, \quad s_n^k := \sin\frac{2\pi k}{n}, \quad c_n := c_n^1, \quad \omega_n := 16\lambda_n - 4,$$

$$\lambda_n := \frac{1}{16}(c_n + 5 + \sqrt{(c_n+9)(c_n+1)}), \sigma_n := \begin{cases} 0.53 & \text{if } n = 3, \\ \frac{1}{4\lambda_n} & \text{if } n > 3, \end{cases}$$

$$\delta_1 := \omega_n c_n^l, \quad \beta_1 := c_n^l + c_n^{l+1}, \quad \delta_2 := \omega_n s_n^l, \quad \beta_2 := s_n^l + s_n^{l+1}.$$

Both for $n^0 = 4$ and $n^0 \neq 4$, $\mathbf{b}_{11}^k$ is placed according to the $1:1:1$ split of knot insertion: $\mathbf{b}_{11}^k := \frac{6(\mathbf{b}_{10}^k + \mathbf{b}_{01}^k) - 4\mathbf{b}_{00}^k + \mathbf{q}_{11}^k}{9}$.

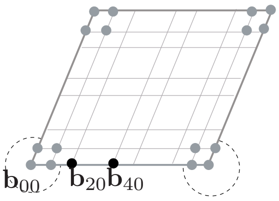
2.6. **Step 3.** *determine the boundary*

3a. [**if** $n^0 = 4 = n^k$] skip Steps 3 and 4.

3b. [**if** $n^0 \neq 4 \neq n^k$] We choose $\alpha_i^k(u) := \lambda_i^k(1-u) + \lambda_{i+1}^k u$, $i = 0, 1, 2$, in (2) and

$$\lambda_0^k := 2\cos(\frac{2\pi}{n^0}), \quad \lambda_3^k := -2\cos(\frac{2\pi}{n^k}), \quad \lambda_1^k := \frac{2\lambda_0^k + \lambda_3^k}{3}, \lambda_2^k := \frac{\lambda_0^k + 2\lambda_3^k}{3}. \quad (8)$$

Therefore, we replace $\mathbf{b}_{20}^k$ and $\mathbf{b}_{70}^k$ by



$$\mathbf{b}_{20}^k := \mathbf{b}_{10}^k + \frac{3(\mathbf{b}_{11}^k + \mathbf{b}_{11}^{k-1} - 2\mathbf{b}_{10}^k) - \lambda_1^k(\mathbf{b}_{10}^k - \mathbf{b}_{00}^k)}{2\lambda_0^k},$$

and a new $\mathbf{b}_{70}^k$ computed symmetrically to $\mathbf{b}_{20}^k$.

Using the re-computed $\mathbf{b}_{20}^k$ and $\mathbf{b}_{70}^k$,

$\mathbf{b}_{40}^k := \frac{4}{3}\mathbf{b}_{20}^k - \frac{1}{3}\mathbf{b}_{80}^k + \frac{2}{3}\mathbf{b}_{70}^k - \frac{2}{3}\mathbf{b}_{10}^k$, and

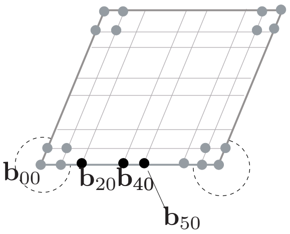
$\mathbf{b}_{50}^k$ is computed symmetrically to $\mathbf{b}_{40}^k$.

3c. [**if** $n^0 \neq 4 = n^k$] By [PF09], not all α_i^k can be linear. We choose

$$\alpha_0^k(u) := \lambda_0^k(1-u) + \lambda_1^k u, \quad \alpha_1^k(u) := \lambda_1^k(1-u)^2 + \lambda_2^k(1-u)u + \lambda_3^k u^2, \quad \alpha_2^k(u) = \lambda_3^k \quad (9)$$

and set $\lambda_0^k := 2\cos(\frac{2\pi}{n^0})$, $\quad \lambda_1^k := \frac{\lambda_0^k}{2}$, $\quad \lambda_2^k := 0$, $\quad \lambda_3^k := 0$.

Then we replace (overwrite) $\mathbf{b}_{20}^k$, $\mathbf{b}_{40}^k$ and $\mathbf{b}_{50}^k$ by



$$\mathbf{b}_{20}^k := \mathbf{b}_{10}^k + \frac{3(\mathbf{b}_{11}^k + \mathbf{b}_{11}^{k-1} - 2\mathbf{b}_{10}^k) - \lambda_1^k(\mathbf{b}_{10}^k - \mathbf{b}_{00}^k)}{2\lambda_0^k},$$

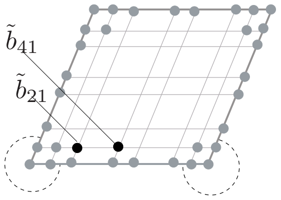
$$\mathbf{b}_{40}^k := \frac{41}{25}\mathbf{b}_{20}^k + \frac{4}{25}\mathbf{b}_{70}^k - \frac{4}{5}\mathbf{b}_{10}^k,$$

$$\mathbf{b}_{50}^k := \frac{36}{25}\mathbf{b}_{20}^k + \frac{9}{25}\mathbf{b}_{70}^k - \frac{4}{5}\mathbf{b}_{10}^k.$$

Note that $\mathbf{b}_{70}^k$ not replaced but stays unchanged from knot insertion. This will aid in guaranteeing C^2 continuity at regular vertices, where $n = 4$.

2.7. **Step 4.** *First interior layer, enforcing G^1 constraints*

4a. [**if** $n^0 \neq 4$ **or** $n^k \neq 4$] compute preliminary coefficients $\tilde{b}_{ij}^k$:



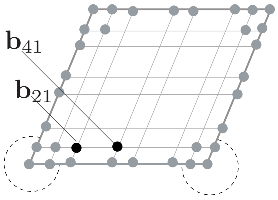
$$\tilde{b}_{21}^k := -\frac{4}{9}\mathbf{b}_{01}^k + \frac{4}{3}\mathbf{b}_{11}^k + \frac{1}{3}\mathbf{b}_{81}^k - \frac{2}{9}\mathbf{b}_{91}^k,$$

$$\tilde{b}_{41}^k := -\frac{20}{27}\mathbf{b}_{01}^k + \frac{4}{3}\mathbf{b}_{11}^k + \mathbf{b}_{81}^k - \frac{16}{27}\mathbf{b}_{91}^k;$$

$\tilde{b}_{51}^k, \tilde{b}_{71}^k$ are computed symmetrically to $\tilde{b}_{41}^k$ and $\tilde{b}_{21}^k$.

Then
4b. [**if** $n^0 \neq 4 \neq n^k$]

$$h_1 := \mathbf{b}_{20}^k + \frac{\lambda_0 \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2} + 2\lambda_1(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k)}{6}, \quad h_2 := \mathbf{b}_{40}^k + \frac{2\lambda_1(\mathbf{b}_{50}^k - \mathbf{b}_{40}^k) + \lambda_2 \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2}}{6},$$



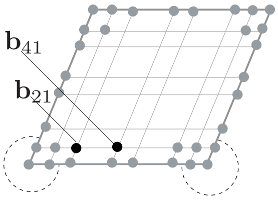
$$\mathbf{b}_{21}^k := h_1 + \frac{1}{2}(\tilde{b}_{21}^k - \tilde{b}_{12}^{k-1}),$$

$$\mathbf{b}_{41}^k := h_2 + \frac{1}{2}(\tilde{b}_{41}^k - \tilde{b}_{14}^{k-1}) \text{ and}$$

$\mathbf{b}_{51}^k, \mathbf{b}_{71}^k$ are computed symmetrically to $\mathbf{b}_{41}^k$ and $\mathbf{b}_{21}^k$.

4c. [**if** $n^0 \neq 4 = n^k$]

$$h_1 := \mathbf{b}_{20}^k + \frac{\lambda_0^k \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2} + \lambda_0^k(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k)}{6}, \quad h_2 := \mathbf{b}_{40}^k + \frac{\lambda_0^k \frac{\mathbf{b}_{70}^k - \mathbf{b}_{50}^k}{2}}{12},$$



$$\mathbf{b}_{21}^k := h_1 + \frac{1}{2}(\tilde{b}_{21}^k - \tilde{b}_{12}^{k-1}),$$

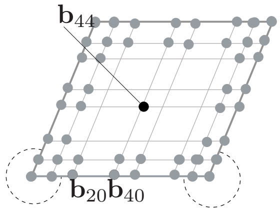
$$\mathbf{b}_{41}^k := h_2 + \frac{1}{2}(\tilde{b}_{41}^k - \tilde{b}_{14}^{k-1}),$$

$$\mathbf{b}_{51}^k := \mathbf{b}_{50}^k + \frac{1}{2}(\tilde{b}_{51}^k - \tilde{b}_{15}^{k-1}).$$

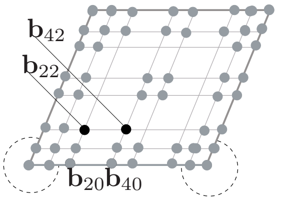
Two layers of control points along each boundary are now fixed. Note that $\mathbf{b}_{71}^k$ remains unchanged from knot insertion. This guarantees C^2 continuity at regular vertices, where $n = 4$.

2.8. **Step 5.** *Interior*

Interior control points $\mathbf{b}_{44}^k$ are computed such that each of the curves $\mathbf{b}(u, t)$ and $\mathbf{b}(t, v)$ with $0 \leq t \leq 1$ can be a single cubic curve subdivided into three pieces and the remaining coefficients as a least deviation thereof.



$$\mathbf{b}_{44}^k := \frac{-\frac{20}{27}\mathbf{b}_{04}^k + \frac{4}{3}\mathbf{b}_{14}^k + \mathbf{b}_{84}^k - \frac{16}{27}\mathbf{b}_{94}^k}{2} + \frac{-\frac{20}{27}\mathbf{b}_{40}^k + \frac{4}{3}\mathbf{b}_{41}^k + \mathbf{b}_{48}^k - \frac{16}{27}\mathbf{b}_{49}^k}{2}.$$



$$\text{if } n^0 \neq 4 \text{ or } n^k \neq 4,\ \mathbf{b}^k_{42} := \frac{1}{2}\mathbf{b}^k_{41} + \mathbf{b}^k_{44} - \frac{1}{2}\mathbf{b}^k_{45},$$

$$\text{if } n^0 \neq 4,\ \mathbf{b}^k_{22} := \frac{\frac{1}{2}\mathbf{b}^k_{12} + \mathbf{b}^k_{42} - \frac{1}{2}\mathbf{b}^k_{52}}{2} + \frac{\frac{1}{2}\mathbf{b}^k_{21} + \mathbf{b}^k_{24} - \frac{1}{2}\mathbf{b}^k_{25}}{2}.$$

Note that for $n^0 = 4 = n^k$ the first three layers of coefficients remained unperturbed, guaranteeing C^2 continuity across regular edges. We have now specified all tensor-product spline coefficients.

2.9. Setting degrees of freedom: interpolation of points and tangents

Our approach, although minimal in the number of knots, offers some degrees of freedom. These were set heuristically as follows. Our default choice of position and tangent plane mimics Catmull-Clark subdivision surfaces. This smoothes out input mesh features. Alternatively, a user can choose $\mathbf{q}^k_{00}$ to be some other point that the surface should *interpolate* (cf. Figure 6 (d)). Also the tangent plane can be set in Step 2 by choosing $\mathbf{e}_1$ and $\mathbf{e}_2$ differently. Moreover, the coefficients $\mathbf{b}_{11}$ are free to choose. One alternative way to set them is to make them depend on the boundary curve as in PCCM [Pet00]. We opted instead for simplicity. Note that, according to [PF09]there is no additional freedom in the choice of the boundary in Step 3. In Step 4, the coefficient pairs $(\mathbf{b}^k_{21}, \mathbf{b}^{k-1}_{12})$ and $(\mathbf{b}^k_{41}, \mathbf{b}^{k-1}_{14})$ each offer one degree of freedom. We set them so as to minimally perturb their positions after knot insertion just as much as is necessary to accommodate the G^1 constraints. Having set the boundary curves and the first derivatives across the boundaries, the remaining B-spline coefficients are all free to choose if we only require (parametric) C^1 continuity internally. It is not always possible to enforce that all remaining second derivatives match across internal boundaries. By default, we chose the interior coefficients to make the transitions as smooth as possible.

3. Smoothness Verification

The polynomial equalities $\partial_2\mathbf{b}^k(u,0) + \partial_1\mathbf{b}^{k-1}(0,u) = \alpha^k(u)\partial_1\mathbf{b}^k(u,0)$ of the unbiased G^1 constraints (2) hold for the kth curve exactly when all spline coefficients are equal. We will show this by expanding the left side, called Left below, and the right side, called Right. We will superscript = signs with the equation or assignment of the algorithm that implies the equality.

We now show this equality starting at the vertices.

3.1. Ordinary quads and ordinary edges

Since Step 1 converts B-spline control points $\mathbf{p}$ to BB control points $\mathbf{q}$ according to (4) ((5) agrees with the formula in (4) for $n = 4$), ordinary quads join C^2. Also, if one or both quads are not ordinary, but both endpoints of the edge have valence $n = 4$, then the spline coefficients up to second order are determined by the conversion followed by knot insertion. Therefore, the surface is C^2 across T-junctions. We summarize this observation as follows.

Lemma 1. *If both endpoints of a boundary curve have valence* 4 *then two patches meeting along the curve join* C^2.

3.2. Extraordinary quads

Next we consider the general case. To verify the G^1 constraints along the shared tensor-product spline patch curves, it is convenient to define

$$\mathbf{b}^k_{3j} := \frac{\mathbf{b}^k_{2j} + \mathbf{b}^k_{4j}}{2}, \quad \mathbf{b}^k_{6j} := \frac{\mathbf{b}^k_{5j} + \mathbf{b}^k_{7j}}{2}, \quad j = 0, \dots 9.$$

This allows us to interpret the $\mathbf{b}^k_{ij}$ as coefficients of a 3×3 arrangement of C^1-connected patches in BB-form. The coefficients of the polynomials $\partial_2 \mathbf{b}^k(u,0)$, $\partial_1 \mathbf{b}^{k-1}(0,u)$, and $\partial_1 \mathbf{b}^k(u,0)$ in (2) are differences of these BB control points:

$$\mathbf{v}^k_i := \mathbf{b}^k_{i1} - \mathbf{b}^k_{i0}, \quad \mathbf{w}^k_i := \mathbf{b}^{k-1}_{1i} - \mathbf{b}^{k-1}_{0i}, \quad \mathbf{u}^k_i := \mathbf{b}^k_{i+1,0} - \mathbf{b}^k_{i0}. \tag{10}$$

(Since (2) has only one variable, u, each difference has one subscript.) The G^1 constraint (2) is then equivalent to the 8 equations

$$\mathbf{v}^k_0 + \mathbf{w}^k_0 = \lambda^k_0 \mathbf{u}^k_0 \tag{11}$$

$$3(\mathbf{v}^k_\mu + \mathbf{w}^k_\mu) = 2\lambda^k_\ell \mathbf{u}^k_\mu + \lambda^k_{\ell+1} \mathbf{u}^k_{\mu-1}, \quad \mu(\ell) = 1, 4, 8 \tag{12}$$

$$3(\mathbf{v}^k_\mu + \mathbf{w}^k_\mu) = \lambda^k_\ell \mathbf{u}^k_\mu + 2\lambda^k_{\ell+1} \mathbf{u}^k_{\mu-1}, \quad \mu(\ell) = 2, 5, 7 \tag{13}$$

$$\mathbf{v}^k_9 + \mathbf{w}^k_9 = \lambda^k_3 \mathbf{u}^k_8 \tag{14}$$

3.2.1. Expansion at vertices of Extraordinary quads

We first focus again on smoothness at the vertices.

Lemma 2. *The* G^1 *constraint holds at* $\mathbf{q}_{00}$.

Proof We verify the G^1 constraint (11) at $u = 0$ by separately expanding the left side Left and its right side Right. Let $n = n^0 \geq 3$, the valence at $\mathbf{q}_{00}$.

$$\begin{aligned} \text{Left}: \quad & \mathbf{v}^k_0 + \mathbf{w}^k_0 \overset{(10)}{=} \mathbf{b}^{k+1}_{10} - \mathbf{b}_{00} + \mathbf{b}^{k-1}_{10} - \mathbf{b}_{00} \\ & \overset{\textbf{Step2.}}{=} \frac{\mathbf{e}_1(c^{k+1}_n + c^{k-1}_n) + \mathbf{e}_2(s^{k+1}_n + s^{k-1}_n)}{3}. \\ \text{Right}: \quad & \lambda^k_0 \mathbf{u}^k_0 \overset{(10)}{=} \lambda^k_0(\mathbf{b}^k_{10} - \mathbf{b}_{00}) \\ & \overset{\textbf{Step2.}}{=} \lambda^k_0(\mathbf{e}_1 c^k_n + \mathbf{e}_2 s^k_n) \overset{(8),(9)}{=} \frac{2c^1_n c^k_n \mathbf{e}_1 + 2c^1_n s^k_n \mathbf{e}_2}{3}. \end{aligned}$$

Equality of Left and Right for all valences including $n^0 = 4$ follows by standard trigonometric expansion. |||

3.2.2. Smoothness across patch boundaries

For each edge, we now analyze the three remaining valence combinations. By Lemma 1, if $n^0 = 4 = n^k$ then (2) holds with $\alpha^k \equiv 0$.

Lemma 3. *If $n^0 \neq 4 \neq n^k$ then* (2) *holds.*

Proof By Lemma 2 and symmetry of the constraint structure at the two endpoints, we need only verify (12) for $\ell = 0$ and $\ell = 1$ and (13) for $\ell = 0$.
For $\ell = 0$, $\mu(\ell) = 1$, (12) has a left hand side, Left, and a right hand side, Right, that agree:

$$\begin{aligned}
\text{Left}: \quad & 3(\mathbf{v}_1^k + \mathbf{w}_1^k) \overset{(10)}{=} 3(\mathbf{b}_{11}^k - \mathbf{b}_{10}^k + \mathbf{b}_{11}^{k-1} - \mathbf{b}_{01}^{k-1}).\\
\text{Right}: \quad & 2\lambda_0^k \mathbf{u}_1^k + \lambda_1^k \mathbf{u}_0^k \overset{(10)}{=} 2\lambda_0^k(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k) + \lambda_1^k(\mathbf{b}_{10}^k - \mathbf{b}_{00})\\
& \overset{\mathbf{3b.}}{=} 2\lambda_0^k\Big(\mathbf{b}_{10}^k + \frac{3(\mathbf{b}_{11}^k + \mathbf{b}_{11}^{k-1} - 2\mathbf{b}_{10}^k) - \lambda_1^k(\mathbf{b}_{10}^k - \mathbf{b}_{00})}{2\lambda_0^k} - \mathbf{b}_{10}^k\Big) + \lambda_1^k(\mathbf{b}_{10}^k - \mathbf{b}_{00})\\
& = 3(\mathbf{b}_{11}^k - \mathbf{b}_{10}^k + \mathbf{b}_{11}^{k-1} - \mathbf{b}_{01}^{k-1}).
\end{aligned}$$

For $\ell = 1, \mu(\ell) = 4$, (12) holds since

$$\begin{aligned}
\text{Left}: \quad & 3(\mathbf{v}_4^k + \mathbf{w}_4^k) \overset{(10)}{=} 3(\mathbf{b}_{41}^k - \mathbf{b}_{40}^k + \mathbf{b}_{14}^{k-1} - \mathbf{b}_{04}^{k-1})\\
& \overset{\mathbf{4b.}}{=} 3\Big(h_2 + \frac{1}{2}(\tilde{b}_{41}^k - \tilde{b}_{14}^{k-1}) - \mathbf{b}_{40}^k + h_2 + \frac{1}{2}(\tilde{b}_{14}^k - \tilde{b}_{41}^{k-1}) - \mathbf{b}_{04}^{k-1}\Big)\\
& = 6(h_2 - \mathbf{b}_{40}^k) \overset{\mathbf{4b.}}{=} 6\Big(\mathbf{b}_{40}^k + \frac{2\lambda_1(\mathbf{b}_{50}^k - \mathbf{b}_{40}^k) + \lambda_2 \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2}}{6} - \mathbf{b}_{40}^k\Big)\\
& = 2\lambda_1(\mathbf{b}_{50}^k - \mathbf{b}_{40}^k) + \lambda_2 \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2}.\\
\text{Right}: \quad & 2\lambda_1^k \mathbf{u}_4^k + \lambda_2^k \mathbf{u}_3^k \overset{(10)}{=} 2\lambda_1^k(\mathbf{b}_{50}^k - \mathbf{b}_{40}^k) + \lambda_2^k(\mathbf{b}_{40}^k - \mathbf{b}_{30}^k).
\end{aligned}$$

The two expressions are equal because $\mathbf{b}_{30}^k = \frac{\mathbf{b}_{40}^k + \mathbf{b}_{20}^k}{2}$.
We verify (13) for $\ell = 0, \mu(\ell) = 2$,

$$\begin{aligned}
\text{Left}: \quad & 3(\mathbf{v}_2^k + \mathbf{w}_2^k) \overset{(10)}{=} 3(\mathbf{b}_{21}^k - \mathbf{b}_{20}^k + \mathbf{b}_{12}^{k-1} - \mathbf{b}_{02}^{k-1})\\
& \overset{\mathbf{4b.}}{=} 3\Big(h_1 + \frac{1}{2}(\tilde{b}_{21}^k - \tilde{b}_{12}^{k-1}) - \mathbf{b}_{20}^k + h_1 + \frac{1}{2}(\tilde{b}_{12}^{k-1} - \tilde{b}_{21}^k) - \mathbf{b}_{02}^{k-1}\Big)\\
& = 6(h_1 - \mathbf{b}_{20}^k) \overset{\mathbf{4b.}}{=} 6\Big(\mathbf{b}_{20}^k + \frac{\lambda_0 \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2} + 2\lambda_1(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k)}{6} - \mathbf{b}_{20}^k\Big)\\
& = \lambda_0 \frac{\mathbf{b}_{40}^k - \mathbf{b}_{20}^k}{2} + 2\lambda_1(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k) = \lambda_0(\mathbf{b}_{30}^k - \mathbf{b}_{20}^k) + 2\lambda_1(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k).\\
\text{Right}: \quad & \lambda_0^k \mathbf{u}_2^k + 2\lambda_1^k \mathbf{u}_1^k \overset{(10)}{=} \lambda_0(\mathbf{b}_{30}^k - \mathbf{b}_{20}^k) + 2\lambda_1(\mathbf{b}_{20}^k - \mathbf{b}_{10}^k).
\end{aligned}$$

This completes the proof. |||

Lemma 4. *If $n^0 \neq 4 = n^k$ then* (2) *holds.*

Proof For the leftmost segment, $\ell = 0$, (11) holds by Lemma 2; and (12) and (13) have already been verified in Lemma 3. For the rightmost segment, $\ell = 2$, Lemma 1 applies. We need only verify (2) for the middle segment, $\ell = 1$. By the formulas in **3c.**, the third derivative of the middle boundary curve segment is

$$\begin{aligned}\mathbf{b}_{30}^k - 3\mathbf{b}_{40}^k + 3\mathbf{b}_{50}^k - \mathbf{b}_{60}^k &= \frac{\mathbf{b}_{20}^k + \mathbf{b}_{40}^k}{2} - 3\mathbf{b}_{40}^k + 3\mathbf{b}_{50}^k - \frac{\mathbf{b}_{50}^k + \mathbf{b}_{70}^k}{2} \\ &= \frac{\mathbf{b}_{20}^k - 5(\frac{41}{25}\mathbf{b}_{20}^k + \frac{4}{25}\mathbf{b}_{70}^k - \frac{4}{5}\mathbf{b}_{10}^k) + 5(\frac{36}{25}\mathbf{b}_{20}^k + \frac{9}{25}\mathbf{b}_{70}^k - \frac{4}{5}\mathbf{b}_{10}^k) - \mathbf{b}_{70}^k}{2} \\ &= 0. \end{aligned} \tag{15}$$

That is, this segment is of degree 2 and can equivalently be defined by coefficients $b_{j0}^{[2]}$ of the polynomial in quadratic BB form:

$$b_{30}^{k[2]} := \mathbf{b}_{30}^k, \quad b_{40}^{k[2]} := \frac{3\mathbf{b}_{40}^k - \mathbf{b}_{30}^k}{2} = \frac{3\mathbf{b}_{50}^k - \mathbf{b}_{60}^k}{2}, \quad b_{50}^{k[2]} := \mathbf{b}_{60}^k. \tag{16}$$

By (9), $\alpha_1^k(u) = \lambda_1^k(1-u)^2$ is a quadratic function and $\lambda_2^k = \lambda_3^k = 0$ (λ_0^k and λ_1^k are given near (9)). For $\ell = 1$, (12) and (13) specialize to

$$9(\mathbf{v}_\mu^k + \mathbf{w}_\mu^k) = \qquad 2\lambda_\ell^k \mathbf{u}_\mu^{k[2]}, \quad \mu(\ell) = 4 \tag{17}$$

$$3(\mathbf{v}_\mu^k + \mathbf{w}_\mu^k) = \qquad 0, \quad \mu(\ell) = 5, \tag{18}$$

where $\mathbf{u}_\mu^{k[2]} = b_{\mu+1,0}^{k[2]} - b_{\mu 0}^{k[2]}$.
We verify (17),

$$\text{Left}: \quad 9(\mathbf{v}_4^k + \mathbf{w}_4^k) \overset{\mathbf{4c.}}{=} 18(h_2 - \mathbf{b}_{40}^k) = 18(\frac{\lambda_0^k \frac{\mathbf{b}_{70}^k - \mathbf{b}_{50}^k}{2}}{12}) = \frac{3\lambda_0^k(\mathbf{b}_{70}^k - \mathbf{b}_{50}^k)}{4},$$

$$\text{Right}: \quad 2\lambda_1^k \mathbf{u}_4^{k[2]} = \lambda_0^k(b_{50}^{k[2]} - b_{40}^{k[2]}) \overset{(16)}{=} \frac{3\lambda_0^k(\mathbf{b}_{60}^k - \mathbf{b}_{50}^k)}{2} = \frac{3\lambda_0^k(\mathbf{b}_{70}^k - \mathbf{b}_{50}^k)}{4}$$

and (18),

$$\begin{aligned}\text{Left}: \quad 3(\mathbf{v}_5^k + \mathbf{w}_5^k) &\overset{\mathbf{4c.}}{=} 3(\mathbf{b}_{50}^k + \frac{1}{2}(\tilde{b}_{51}^k - \tilde{b}_{15}^{k-1}) + \mathbf{b}_{50}^k + \frac{1}{2}(\tilde{b}_{15}^{k-1} - \tilde{b}_{51}^k) - 2\mathbf{b}_{50}^k) \\ &= 0.\end{aligned}$$

This completes the proof. |||

3.3. Overall Smoothness

Since the tensor-product spline patches are of degree bi-3 and have at most double internal knots, they are internally parametrically C^1. Together with Lemma 3 and Lemma 4 this implies the claimed smoothness. In particular, by Lemma 1, ordinary patches join C^2 with extraordinary patches. We summarize.

Theorem 1. *A surface generated by the algorithm is C^1; and it is C^2 across regular edges.*

4. Examples

In a large model, such as Figure 7 and Figure 1, an overall shape impression dominates. To test the shape of the surface construction and illustrate its proven properties, we therefore focus intentionally on small synthetic examples where the shape can be predicted.

Figure 4, *top*, illustrates that the construction is well-behaved for some geometrically un-symmetric configurations of valence 3,4,5 and 6. We chose un-symmetric configurations since geometrically symmetric input can hide continuity defects. The neighborhood of the vertex of valence 6 a form of 'monkey saddle'. We do not see the shape deficiency, a flat region, that according to [PF09]*any* bi-3 construction with quadratic α at the vertices (e.g. PCCM, [GZ94, vW86]) must have. The refinement of the quad mesh in Figure 4, *bottom*, illustrates the localized nature of the 3×3 split near extraordinary points: (e) and (g) show the corresponding $\mathbf{q}$ and $\mathbf{b}$ patches for regular and irregular quads. That is, each (randomly chosen) color corresponds to one polynomial piece (cf. Figure 1). In tensor-product regions, the construction yields just a standard bi-3 tensor-product C^2 spline surface. Figure 5 allows us to look in more detail at the transition between standard bi-3 C^2 tensor-product splines (center region of the lozenge-shaped test example) and the construction for an irregular quad. Despite the T-corners, the Gaussian curvature shading attests to the C^2 continuity of the transition. This is trivially so since the construction simply subdivides, and then (cf. Steps 1–5) does not perturb the control net of the three sub-patches in the 3×3 arrangement bordering on the standard bi-2 C^2 tensor-product spline. The T-junctions present no problem since the surfaces are mathematically C^2. For example, if we evaluate with uniform parameter spacing, we can simply evaluate the standard bi-3 C^2 tensor-product splines three times more finely to match points. Figure 5 (d) and (e) allow us to compare the current construction and the construction according to [FP08]. Juxtaposing Gaussian curvature shading, we notice that the construction of the present paper improves the transition between regular and irregular quads. Figure 6, represents not only a basic check, but illustrates in (d) the option to locally interpolate the quad net by setting $\mathbf{q}_{00}$ to the vertex position (see Section 2.4).

5. Discussion and Extensions

In applications such as surface reconstruction, many more knots and therefore polynomial pieces per quad may be needed to fit given data. By inserting knots in the bi-3 tensor-product spline patch of this paper, we can refine the representation of the patch. All interior coefficients are then freely adjustable to fit the data. For example, by augmenting κ_2 to $(0,0,0,0,1,1,2,3,4,5,6,7,8,8,9,9,9,9)$, all coefficients with Greville abscissae i, j, $1 < i, j < 8$, can be freely chosen. We note that according to [PF09, Thm 1]inserting *single* knots in a bi-3 construction does not add degrees of freedom to the *boundary curves*. This is a general and not a specific restriction of the present construction.

For some combinations of vertex valences, the number of knots could be lowered still to just one internal knot. For example, when $n^0 = n^1$ on opposing edges then one internal knot suffices to construct a smoothly connect patch. However, this would create many special rules, or severely restrict the class of input meshes [PF10]. But minimizing the number of knots may not be the main argument for using the present algorithm over other bi-3 constructions. The main improvement of the presented algorithm over [Pet00, GZ94, vW86] is not the lower number of polynomial pieces, but the improved shape for higher-order saddles. The key to this is the reparameterization at the vertices: both α_0^k and α_2^k in (8) or (9) are linear, allowing for degree 3 boundary curves emanating from the vertex. By comparison [Pet00, GZ94, vW86] use quadratic α_0^k and α_2^k to quickly switch from G^1 constraints to C^2 continuity across the boundaries. At higher-order saddle-points, using quadratic α_0^k for a bi-3 construction will lead to flat spots [PF09, Lemma 5]. So, the Ansatz in [Pet00], [GZ94, (4.3)] or [vW86] cannot always yield good shape at higher-order saddle-points. The new algorithm does not suffer from this source of shape deficiency.

We note that all coefficients $\mathbf{q}_{ij}$ are convex combinations of the input data; and all coefficients $\mathbf{b}_{ij}$ are at least affine combinations: that is, on setting the input data to 1, the formulas output 1. Note for example, that in Step 2, $\sum_{l=1}^{n} \delta_1 = \sum_{l=1}^{n} \omega_n c_n^l = 0$ and similarly for the sum of the δ_2 and of β_i, for $i = 1, 2$.

6. Conclusion

We presented explicit formulas for constructing the B-spline coefficients of a C^1 surface. The formulas are intended to make it easy to implement the surface construction and reproduce the results. We know to date of one commercial package that has adopted this construction for extraordinary points. Vertex positions and tangents are either set, by default, approximating Catmull-Clark subdivision surfaces, or they can be set to match existing data. The latter reminds of Catmull-Rom splines. While the underlying machinery is not new but classic, the new construction deserves attention since, among bicubic spline surface constructions with one spline patch per quad, it uses the fewest number and multiplicity of internal knots (hence polynomial pieces) to allow for an unbiased G^1 construction with n-valent vertices without flat regions near higher-order saddles.

References

[Bez77] Pierre E. Bezier. *Essai de definition numerique des courbes et des surfaces experimentales.* Ph.d. thesis, Universite Pierre et Marie Curie, February 1977.

[CC78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.

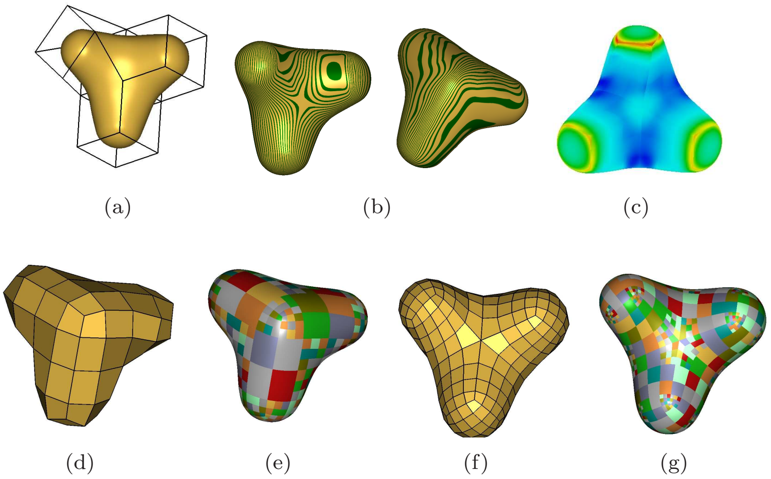


Figure 4: **Convex and saddle shapes; localized bi-3 refinement.** (*top*) (a) quad net and surface: vertices of valence 3,4,5 and 6 with non-symmetric neighborhoods; valence 6, in the center, corresponds to an asymmetric 'monkey saddle'; (b) highlight lines, (c) Gaussian curvature shading of the resulting surface. (*bottom*) (d) quad net obtained from (a) by one step of Catmull-Clark subdivision, (f) quad net obtained from (a) by two steps of Catmull-Clark subdivision; (e) and (g) show the corresponding **q** patches for regular and **b** patches for irregular quads.

[FP08] Jianhua Fan and Jörg Peters. On smooth bicubic surfaces from quad meshes. In G. Bebis et al., editor, *ISVC (1)*, volume 5358 of *Lecture Notes in Computer Science*, pages 87–96. Springer, 2008.

[Gre74] J. A. Gregory. *Smooth interpolation without twist constraints*, pages 71–88. Academic Press, 1974.

[GZ94] John A. Gregory and Jianwei Zhou. Filling polygonal holes with bicubic patches. *Comput. Aided Geom. Des.*, 11(4):391–410, 1994.

[LS08] Charles Loop and Scott Schaefer. Approximating Catmull-Clark subdivision surfaces with bicubic patches. *ACM Trans. Graph.*, 27(1):1–11, 2008.

[MYP08] A. Myles, Y. Yeo, and J. Peters. GPU conversion of quad meshes to smooth surfaces. In D. Manocha, B. Levy, and H. Suzuki, editors, *ACM Solid and Physical Modeling Symposium, June 2 - 4, 2008,Stony Brook University, Stony Brook, New York, USA*, pages 321–326. ACM Press, 2008.

[Pet91] J. Peters. Smooth interpolation of a mesh of curves. *Constructive Approximation*, 7:221–247, 1991. Winner of SIAM Student Paper Competition 1989.

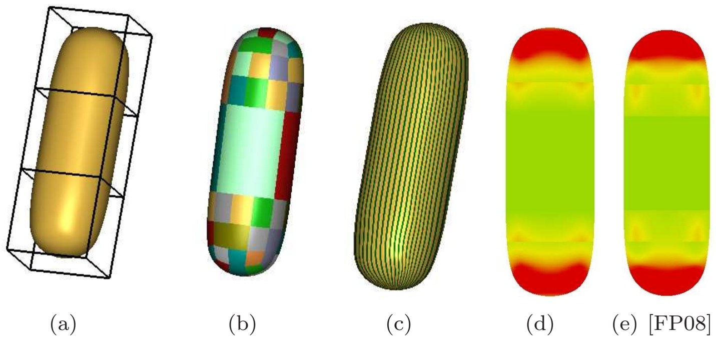


Figure 5: C^2 **transition between regular and irregular quads and comparison to [FP08]** (a) quad net and surface; (b) partition into regular bi-3 C^2 tensor-product splines (central patches) $\mathbf{q}$ for regular quads and composite 3×3 split patches $\mathbf{b}$ patches for irregular quads; (c) highlight lines; (d) and (e) juxtaposition of the Gaussian curvature shading of the presented construction (a,b,c) and the construction according to [FP08].

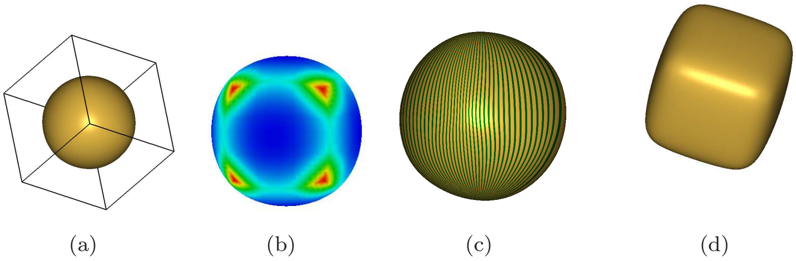


Figure 6: **Approximation and interpolation.** (a) quad net and surface; (b) Gaussian curvature shading; (c) highlight lines. (d) Interpolation of the quad net (see Section 2.4).

[Pet00] J. Peters. Patching Catmull-Clark meshes. In K. Akeley, editor, *ACM Siggraph*, pages 255–258, 2000.

[Pet01] J. Peters. Modifications of PCCM. Technical Report 2001-001, Dept CISE, University of Florida, 2001.

[PF09] J. Peters and Jianhua Fan. On the complexity of smooth spline surfaces from quad meshes. *Computer-Aided Geometric Design*, 27:96–105, 2009. doi:10.1016/j.cagd.2009.09.003.

[PF10] J. Peters and J. Fan. The projective linear transition map for constructing smooth surfaces. In Jean-Philippe Pernot, editor, *SMI 10 conference*, pages xx–xx. IEEE, June 21-23 2010.

[vW86] J. van Wijk. Bicubic patches for approximating non-rectangular control-point meshes. *Computer Aided Geometric Design*, 3(1):1–13, 1986.

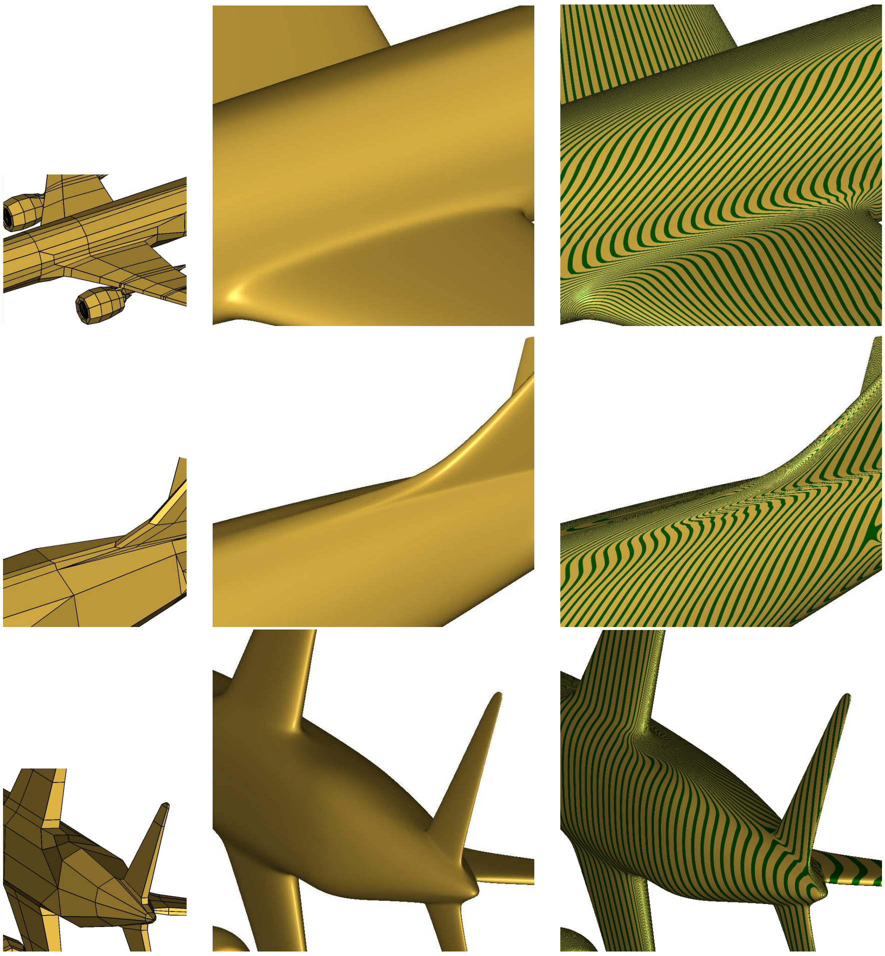


Figure 7: Views of the airplane model in Figure 1: from *left* to *right*: quad mesh, shaded, highlight lines.

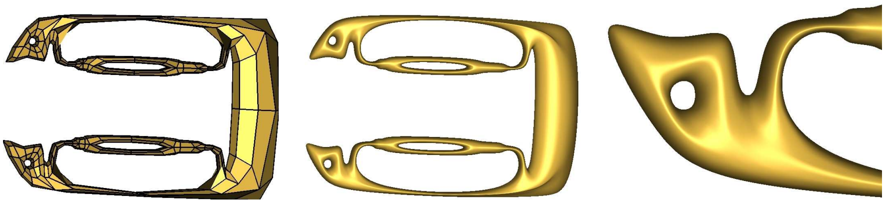


Figure 8: From *left* to *right*: quad mesh, shaded, zoom view.