Biquartic C^1-surface splines over irregular meshes

by

Jörg Peters[†]
jorg@cs.purdue.edu

Key words: C^1 surface, corner cutting, tensor-product splines, spline mesh, blending, vertex-degree, polyhedral

Running title: Biquartic C^1-surface splines

Version: Jan 95 **Date printed:** January 26, 1995

Submitted to: CAD

Abstract

C^1-surface splines define tangent continuous surfaces from control points in the manner of tensor-product (B-)splines, but allow a wider class of control meshes capable of outlining arbitrary free-form surfaces with or without boundary. In particular, irregular meshes with non quadrilateral cells and more or fewer than four cells meeting at a point can be input and are treated in the same conceptual frame work as tensor-product B-splines; that is, the mesh points serve as control points of a smooth piecewise polynomial surface representation that is local and evaluates by averaging.

Biquartic surface splines extend and complement the definition of C^1-surface splines in [Peters '93], improving continuity and shape properties in the case where the user chooses to model entirely with four-sided patches. While tangent continuity is guaranteed, it is shown that no polynomial, symmetry-preserving construction with adjustable blends can guarantee its surfaces to lie in the local convex hull of the control mesh for very small blends where three patches join. Biquartic C^1-surface splines do as well as possible by guaranteeing the property whenever more than three patches join and whenever the blend exceeds a certain small threshold.

[†] Department of Computer Science, Purdue University, W-Lafayette IN 47907-1398
Supported by NSF grant CCR-9211322

1. Introduction

A single tensor-product B-spline complex can only model a small subclass of surfaces that arise in geometric modeling, because every interior point of the B-spline control mesh must be surrounded by exactly four quadrilateral cells. Thus the surface pieces must form a regular, checker board arrangement that allows only deformations of the plane and deformations of the torus to be modeled without singularity. Even for such topologically restricted objects, the rigid structure prevents a natural modeling of frequent features such as suitcase corners or house corners, where three or five quadrilaterals meet. In practice, patch trimming and certain singular parametrizations are used to overcome the restrictions. However, by destroying the consistent B-spline framework, these techniques create a number of special cases and difficult problems such as desingularization and the need for smoothly joining trimmed surfaces.

Tangent plane continuous splines over irregular meshes defined in [Peters '93] overcome the rigid tensor-product frame work, allowing freedom both in the number of patches meeting at a vertex and the number of edges to a mesh cell while preserving the desirable properties of B-splines, such as a low degree polynomial or rational representation of maximal smoothness and a geometrically intuitive variation of the surface in terms of the coefficients as evidenced by a local convex hull property. The local convex hull property, i.e. the property that every point on the surface is a convex combination of the mesh points nearby, also implies that linear features can be reproduced, for example faces and parts of faces of an input polyhedron can be interpolated, while shape handles in the form of blend ratios give direct control over blend radii. A more detailed list of properties of surface splines is compiled at the end of this section. Splines over irregular meshes come in three flavors subject to user choice: surfaces represented entirely by three-sided patches of degree three that reduce to quadratic box-splines at regular mesh sites, surfaces represented entirely by bicubic, four-sided patches that reduce to biquadratic tensor-product splines at regular mesh sites and a recommended mixed representation that uses biquadratic tensor-product splines at regular sites and three-sided cubics at irregular mesh sites. Only the representation in terms of just bicubic, four-sided patches does not fully meet the criteria of a surface spline. There are input mesh configurations for which the error in the continuity constraints is only minimized, but the resulting surface is not strictly C^1. (Here I follow the established convention of differential geometry that a surface is C^1 if the maps that parametrize it join once continuously differentiable after reparametrization; this is consistent with the computer aided design literature which uses the abbreviation G^1 join instead of 'once continuously differentiable join after reparametrization' and reserves the term C^1 join for the case that the reparametrization is the identity.) Moreover, the convex hull property is not guaranteed to hold for small blend ratios; that is when the user forces extreme changes of curvature along cell boundaries, some parts of the surface may no longer be a convex combination of the input mesh. This violation of the convex hull where three patches surround a vertex is endemic to any piecewise polynomial, symmetry-preserving construction that allows adjusting the blend arbitrarily small. (Surface splines based on three-sided patches do not encounter this problem since they split the mesh cells diagonally; thus at least six patches join at any interior mesh point.) Thus it would be sensible to avoid modeling with just four-sided patches. Yet, many modeling and display

environments restrict the user in this choice by not providing three-sided patch representations. Therefore this paper develops a biquartic surface spline representation that guarantees oriented tangent continuity and comes with a proof of the convex hull property where more than three patches join. Where three patches join, the convex hull property is guaranteed if the blend is allowed to round at least one millionth of the mesh cell (see Proposition 3.4 for the details). As in the general case the blending of one surface feature into another is controlled by shape handles in the form of blend ratios associated with each edge. This yields local control of the change of the normal and the curvature across that edge.

Related work. Building on work by [Sabin '83] and [Goodman '91], [Höllig, Mögerle '89] pioneered the idea of geometrically continuous spline spaces Explicit representations of such G-spline surfaces in terms of say the Bernstein-Bézier form require the solution of a large linear, irregularly sparse system of equations to match data. Newer work ([Piah '91], [Lee, Majid '91] and [Reif '93]), improves by exhibiting explicit formulas for the coefficients of the surface patches in terms of the input mesh. Still, the size of the support of these formulas makes it non trivial to predict the shape of the resulting surface. The formulas for the biquartic G-splines introduced in this paper, in contrast, are simple enough to check, say the convex hull property. Surface splines are also related to the subdivision algorithms [Doo '78] and [Catmull '78] in that their first construction step subdivides the input mesh and creates a refined mesh. Surface splines are based on reparametrization and geometric smoothness as surveyed in [Gregory '90], but differ from approaches like [Sarraga '87], [Hahn '89] and [van Wijk '84] in that no constraint systems have to be solved at run time to enforce patch to patch smoothness. B-patches ([Seidel '91], [Dahmen, Micchelli, Seidel '92]) and S-patches [Loop, DeRose '90] provide alternative approaches to defining smooth surfaces over irregular meshes at the cost of a less commonly available patch representation.

Structure of the paper. Section 2 defines the biquartic C^1-surface splines, Section 3 establishes their continuity, vector space and shape properties and Section 4 lists extensions and shows examples. This section concludes with a list of *properties of surface splines* (cf. [Peters '93]).

- *free-form modeling capability.* There are no restrictions on the number of cells meeting at a mesh point or the number of edges to a mesh cell. Mesh cells need not be planar.
- *built-in smoothness (unless explicitly reduced) and local smoothness preserving editability.* For given connectivity and shape parameters, the surface spline form a vector space of geometrically smooth surface parametrizations. In order to manipulate the surface spline, it suffices to average or move the mesh points locally.
- *low degree parametrization.* The surface is parametrized by low degree polynomial patches. The representation can be extended to rational patches by using a fourth coordinate.
- *simple interpolation.* Mesh points and normals can be interpolated without solving a system of constraints.
- *evaluation by averaging.* The coefficients of the parametrization in Bernstein-Bézier

form can be obtained by applying averaging masks to the input mesh. (The Bernstein-Bézier form in turn is evaluated by averaging.) Thus the algorithm is local and can be interpreted as a rule for cutting an input polytope such that the limit polytope is the spline surface.

- *convex hull property.* The surface lies locally and globally in the convex hull of the input mesh. In other words, every point on the surface can be computed as an average of the mesh points with coefficients that are positive and sum to one (cf. Proposition 3.4)
- *intuitive shape parameters.* The averaging process is geometrically intuitive. Parameters analogous to knot distances govern the depth of the cuts into the polytope outlined by the control mesh. (In concave regions the complement of the polytope is cut.) Smaller cuts result in a surface that follows the input mesh more closely and changes the normal direction more rapidly across the boundary. In the limit this allows adjusting the built-in smoothness, e.g. reducing it to continuity for zero cuts. Discontinuity can be achieved by a change of mesh connectivity.
- *taut interpolation of the control mesh for zero blend ratios.* Cuts of zero depth result in a singular parametrization at the mesh points analogous to singularities of a spline with repeated knots. The continuity of the surface is reduced, but in return the edges of the input mesh are interpolated and the surface is taut, e.g. planar when the mesh cell is planar.

In general any two of the following three properties can be achieved for any input mesh using C^1-surface splines. This is optimal in the sense that, as the reader may check, no surface construction (in fact not even a curve construction) can achieve all three properties simultaneously for all data sets.

1. The convex hull property
2. Tangent continuity
3. Interpolation of the input mesh points

2. Biquartic C^1-surface splines defined

This section gives an algorithm for generating the Bernstein-Bézier coefficients of a biquartic C^1 surface from an irregular mesh of points. Analogous to tensor-product splines, the biquartic C^1 surface spline is defined by

- mesh points $\mathcal{M}$,
- mesh connectivity $\uparrow$,
- blend ratios (knot spacings) a_{ij},

and an evaluation algorithm. Since the evaluation of polynomials in Bernstein-Bézier form is well understood, we express the surface spline in terms of Bernstein-Bézier patches. A biquartic patch p_i in tensor-product Bernstein-Bézier form is a map

$$[0,1] \times [0,1] \mapsto \mathbb{R}^3$$
$$(u,v) \mapsto p_i := \sum_{\substack{k+k'=4 \\ l+l'=4 \\ k,k',l,l' \geq 0}} b_{kl,i} \frac{4}{k'!k!}(1-u)^{k'} u^k \frac{4}{l'!l!}(1-v)^{l'} v^l$$

where $b_{kl,i} \in \mathbb{R}^3$, $k,l \in \{0,1,2,3,4\}$ are the Bernstein-Bézier coefficients, i.e. the control points of the Bézier net (see e.g. [Boehm, Farin, Kahmann '84] or [Farin '90] for a general treatment of the Bernstein-Bézier form and Figure 2.3 for the association of the coefficients with the surface). Note that each p_i is vector-valued, because the control points $b_{kl,i}$ are in $\mathbb{R}^3$. In the following, V_i, C_i, $\tilde{b}_{ij}$ and P_i are vectors, definitions are emphasized by using := rather than = and subscripts for patches surrounding a vertex are interpreted modulo the number of patches.

The process of deriving the coefficients from the mesh points consists of the following steps.

Algorithm $(\mathcal{M}, \uparrow, a_{ij}) \mapsto b_{kl,i}$.

1 Midpoint mesh refinement.
2 Construction of intermediate control points.
3 Construction of the Bernstein-Bézier coefficients.

The *input* to the algorithm is any mesh of points such that at most two cells abut along any edge. The mesh cells need not be planar, and there is no constraint on the number of cells meeting at a vertex. The mesh may model bivariate surfaces with or without boundary and of arbitrary topological genus. To achieve the design intent, each mesh cell should be convex in the sense that there exists a projection of its vertices into a plane such that none of the projected vertices lies in the convex hull of the other projected vertices. For example, a facet with an inner loop should be broken into convex pieces, since the surface is generated by averaging and would cover the hole defined by the inner loop. While a triangulation would always enforce convexity it is usually sufficient and appropriate to add just a few edges using a standard convexification heuristic. The input *parameters* are as follows. Associated with each pair, cell and cell vertex, is a patch p_i and two scalar weights $0 < a_{ij} < 1$, $a^*_{ij} := 1/2$, $j = 1,2$, called *blend ratios*. (Default settings of a parameter

are indicated by the superscript *.) Geometrically, smaller ratios result in a surface that follows the input mesh more closely and changes the normal direction more rapidly close to the mesh edges. The blend ratios are similar to relative knot spacings in that zero ratios cause the mesh edges to be interpolated and the smoothness to decrease. The blend ratios of each cell may be modified independently of each other and of those in other cells. Parameters α and β are associated with each mesh point and determine the width and distance of the tangent plane from the mesh point. The width of the tangent plane at the midpoints of the patch boundaries is governed by the parameter γ.

The *output* is a C^1 surface that follows the outline of the input mesh and consists of no more than $2e$ biquartic pieces, where e is the number of edges of the input mesh.

1. Midpoint mesh refinement. Inserting additional points at centroid of each cell and the midpoint of each edge, we obtain a refined mesh of quadrilateral subcells and two types of points of the refined mesh: vertices of type M that correspond to edge midpoints and therefore have exactly four neighbors, and vertices of type P that correspond to original mesh points and centroids and are surrounded by vertices of type M.

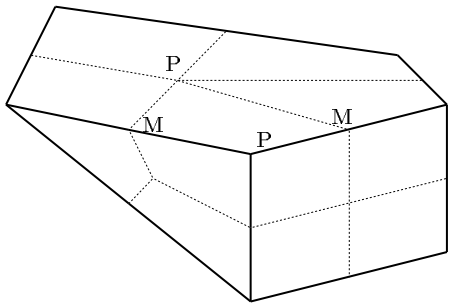

Figure 2.1: Midpoint mesh refinement.

2. Construction of intermediate control points. For each quadrilateral subcell, define a point C_i subject to the blend ratios $0 < a_{i1}, a_{i2} < 1$ as follows. Let V_1, V_2, V_3, V_4 be the vertices of the ith subcell joining at the original input mesh point $P := V_1$. Then

$$C_i := (1-a_{i1})(1-a_{i2})V_1 + (1-a_{i1})a_{i2}V_2 + a_{i1}a_{i2}V_3 + (1-a_{i2})a_{i1}V_4$$

$$P' := (1-\beta)P + \beta \sum_{l=1}^{n} C_l/n \qquad 0 \le \beta \le 1,\ \beta^* := 7/8$$

$$C_i' := P' + \frac{\alpha}{n} \sum_{k=1}^{n} \cos(\frac{2\pi}{n}k) C_{i+k} \qquad 0 \le \alpha \le 1,\ \alpha^* := \beta$$

$$\tilde{b}_{1,i} := (C_i' + C_{i+1}')/2$$

$$\tilde{b}_{2,i} := (C_i + C_{i+1})/2$$

The default choice of α is conservatively small to simplify the proof of the convex hull property: using the default, all points generated so far are convex combinations of the input mesh points. The maximal value of α for which the points obey the local convex hull property depends on the geometry of the input mesh. Choosing $\beta = 0$ yields a surface that interpolates the mesh point. To reduce inflections when $\beta = 0$, the C_i should subsequently be moved into the direction of the average of the normals at the vertices. (The reader may verify that a smooth interpolating surface, regardless of the its construction, is in general not in the local convex hull.)

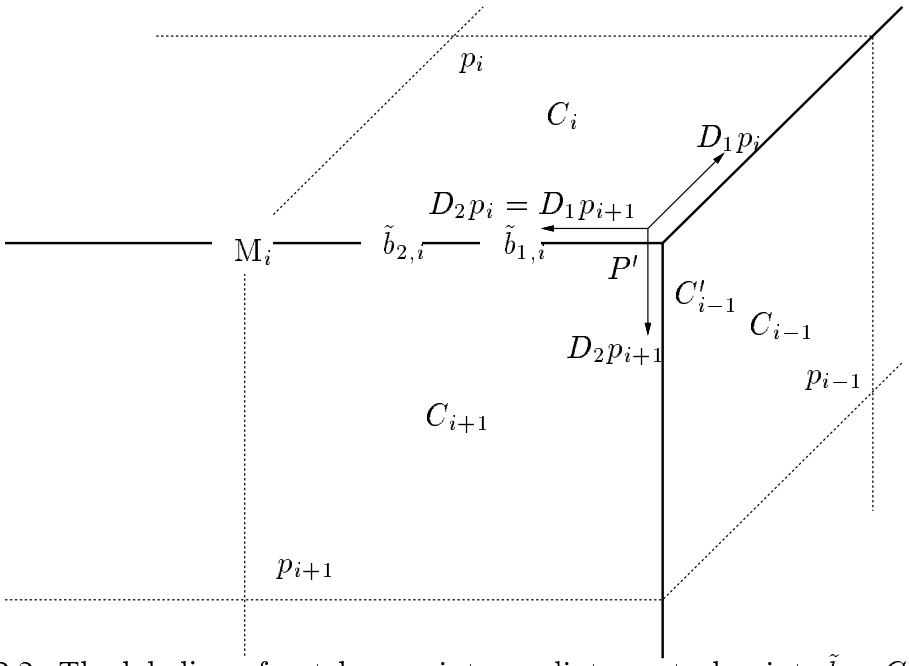

Figure 2.2: The labeling of patches p_i, intermediate control points $\tilde{b}_{ji}$, C_i, C'_i, P' and directional derivatives $D_j p_i$.

For the last step consider a mesh point P and its n neighbors M_i, $i = 1..n$ as illustrated in Figure 2.3. Let $p_i, i = 1..n$ be the biquartic patches in tensor-product Bernstein-Bézier form that surround a common point $p_i(0,0) = b_{00,i}$. The patches are connected so that $p_i(0,t) = p_{i+1}(t,0)$ by setting

$$b_{0\,j,i} = b_{j0,i+1}.$$

The patches are consistently oriented with the parameter t of the curve appearing second in $p_i(0,t)$ but first in $p_{i+1}(t,0)$. To obtain tangent continuity along the common boundary, denote by $D_2 p_i := \lim_{h\to 0}[p_i(u+h,0) - p_i(u,0)]/h$ the versal derivative along the boundary and by $D_1 p_i$ and $D_2 p_{i+1}$ the transversal derivatives along the boundary. The directional derivatives are coplanar by the choice of coefficients in Step 3.

3. Construction of the Bernstein-Bézier coefficients. Let C_{ij}, $j = 1..4$ be the

intermediate control points surrounding M_i. Set

$$b_{00,i} := P'$$
$$b_{01,i} := (3\tilde{b}_{1,i} + b_{00,i})/4$$
$$b_{02,i} := (3\tilde{b}_{2,i} + 3\tilde{b}_{1,i})/6$$
$$b_{03,i} := (3\tilde{b}_{2,i} + b_{04,i})/4$$
$$b_{04,i} := \sum_{l=1}^{4} C_{il}/4$$
$$b_{11,i} := \frac{3c}{8}C_i + \frac{6-3c}{8}C_i' + \frac{1}{4}b_{00,i}$$
$$b_{12,i} := \frac{c}{16}C_{i,j+1} + \frac{8-c}{16}C_i + C_i'/2 - \gamma(C_i - C_{i+1} + C_i' - C_{i+1}'),\ 0 \le \gamma < 1/2,\ \gamma^* := 1/8$$
$$b_{21,i+1} := \frac{c}{16}C_{i,j-2} + \frac{8-c}{16}C_{i+1} + C_{i+1}'/2 + \gamma(C_i - C_{i+1} + C_i' - C_{i+1}')$$
$$b_{13,i} := (3C_i + b_{04,i})/4$$
$$b_{22,i} := C_i$$

All remaining coefficients of the patch p_i are determined by the analogous construction at the other subcell vertex of type P. Tangent continuity of the resulting surface is proved in Theorem 3.1.

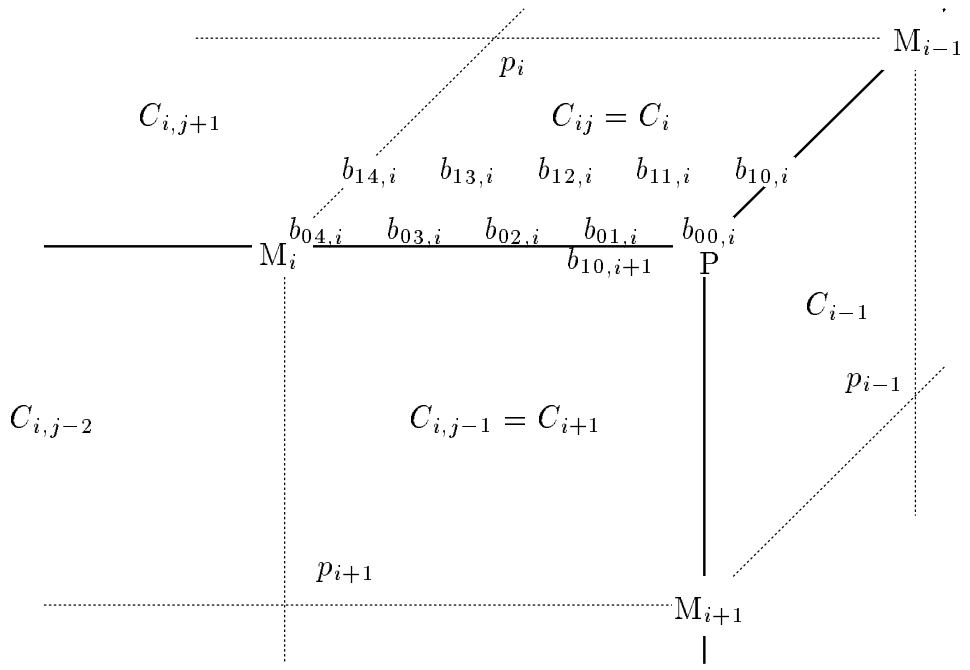

Figure 2.3: Intermediate control points C_i, C_{ij} and two of five layers of Bernstein-Bézier coefficients $b_{jk,i}$ of the patch p_i. Note that $b_{10,i+1} = b_{01,i}$.

3. Smoothness and shape properties

This section shows that the patches generated by the algorithm join G^1 and hence form a tangent continuous, i.e. a C^1 surface. Linearity of differentiation then implies that surface splines with equal blend ratios and connectivity form a vector space. Next we show that the construction generates surfaces with locally adjustable blend such that for the limit value $a_{ij} = 0$, the original mesh is interpolated. While this is desirable, we show subsequently that no symmetry preserving, piecewise polynomial construction with blend adjustable to zero radius can guarantee the convex hull property in the limit at mesh points with three neighbors. That is, not every surface point is an average of the input mesh for certain extreme configurations of data and choice of blend. The biquartic construction is shown to do the best possible. It satisfies the convex hull property whenever the number of neighbors exceeds three or the blend uses at least a small fraction of the available blend surface; in the remaining case, when the number of neighbors is three and the blend is almost a sharp edge, the violation of the convex hull is bounded by 10^{-6} times the distance of the mesh point to its furthest neighbor.

We start with the proof of continuity.

Theorem 3.1. *The biquartic surface splines form a C^1 surface.*

Proof Consider Figures 2.2 and 2.3. By construction of the Bernstein-Bézier coefficients in the previous section, $D_2 p_i$ is a univariate, vector-valued polynomial with coefficients $3(\tilde{b}_{1,i} - b_{00,i})$, $3(\tilde{b}_{2,i} - \tilde{b}_{1,i})$ and $3(b_{04,i} - \tilde{b}_{2,i})$. Since $D_1 p_i$ and $D_2 p_{i+1}$ are of degree 4, the degree count is consistent with the claim that

$$2c(1-t)^2 D_2 p_i = D_1 p_i + D_2 p_{i+1}, \qquad c := \cos(2\pi/n),$$

where n is the number of patches surrounding the vertex of type P. To prove the claim, we need to show that the five coefficients of the quartic polynomials on either side of the equation agree, i.e. that

$$\frac{3}{4} 2c(\tilde{b}_{1,i} - b_{00,i}) = (b_{10,i} + b_{01,i+1} - 2b_{00,i}) \tag{3.0}$$

$$\frac{3}{4} 2c2(\tilde{b}_{2,i} - \tilde{b}_{1,i}) = 4(b_{11,i} + b_{11,i+1} - 2b_{01,i}) \tag{3.1}$$

$$\frac{3}{4} 2c(b_{04,i} - \tilde{b}_{2,i}) = 6(b_{12,i} + b_{21,i+1} - 2b_{02,i}) \tag{3.2}$$

$$0 = 4(b_{13,i} + b_{31,i+1} - 2b_{03,i}) \tag{3.3}$$

$$0 = b_{14,i} + b_{41,i+1} - 2b_{04,i} \tag{3.4}$$

We check the equalities by substitution of the intermediate control points as detailed in Section 2. For example, $b_{10,i} = b_{01,i-1} = (3\tilde{b}_{1,i-1} + b_{00,i})/4 = (3C'_{i-1} + 3C'_i + 2P')/8$, $b_{01,i+1} = (3C'_{i+1} + 3C'_{i+2} + 2P')/8$, hence

$$b_{10,i} + b_{01,i+1} - 2b_{00,i} = \frac{3}{8} \sum_{j=i-1}^{i+2} C'_j - P'$$

while

$$\frac{3}{4}(\tilde{b}_{1,i} - b_{00,i}) = \frac{3}{8}\sum_{j=i}^{i+1} C'_j - P'.$$

Equation (3.0) holds, because it is the sum of two expressions of the type

$$\begin{aligned} & C'_{i-1} - 2cC'_i + C'_{i+1} - 2(1-c)P' \\ &= \frac{\alpha}{n}\sum_{i=1}^{n} C_i\big(\cos(\frac{2\pi}{n}(i-1)) + \cos(\frac{2\pi}{n}(i+1)) - 2\cos(\frac{2\pi}{n})\cos(\frac{2\pi}{n}i)\big) \end{aligned}$$

and $\cos(\frac{2\pi}{n}(i-1)) + \cos(\frac{2\pi}{n}(i+1)) - 2\cos(\frac{2\pi}{n})\cos(\frac{2\pi}{n}i) = 0$. Equation (3.1) can be rewritten and verified as

$$b_{11,i} + b_{11,i+1} = \frac{3c}{8}(C_i + C_{i+1} - C'_i - C'_{i+1}) + 3(C'_i + C'_{i+1})/4 + 2b_{00,i}/4$$

and Equation (3.2) as

$$\begin{aligned} b_{12,i} + b_{21,i+1} &= \frac{c}{16}(\sum_{l=1}^{n} C_l - 2C_i - 2C_{i+1}) + \tilde{b}_{2,i} + \tilde{b}_{1,i} \\ &= \frac{c}{16}(C_{i,j+1} + C_{i,j-2}) + \frac{8-c}{16}(C_i + C_{i+1}) + (C'_i + C'_{i+1})/2. \end{aligned}$$

For the last two equations, let the ith patch at $b_{00,i}$ be the jth patch joining at $b_{04,i}$ as illustrated in Figure 2.3. Then (3.3) equals

$$b_{13,i} + b_{31,i+1} - 2b_{03,i} = 3(C_i + C_{i+1})/4 - 6\tilde{b}_{2,i}/4 = 0$$

and (3.4) reads

$$\begin{aligned} (b_{14,i} - b_{04,i}) + (b_{41,i+1} - b_{04,i}) &= \frac{3}{4}(\tilde{b}_{2,j+1} - b_{04,i}) + \frac{3}{4}(\tilde{b}_{2,j-1} - b_{04,i}) \\ &= \frac{3}{16}(2C_{ij} + 2C_{i,j+1} - \sum_l C_{il} + 2C_{i,j-1} + 2C_{i,j-2} - \sum_l C_{il}) \\ &= 0. \end{aligned}$$

■

If $n = 4$, a similar substitution shows that for appropriate choice of α, β and γ, the surface is curvature continuous at the mesh points of type P, but not necessarily at the points of type M. We now consider the locally adjustable blend in the limit as the blend ratio, and hence the part of the surface over which the normal changes rapidly, decreases to zero.

Proposition 3.2. *If all blend ratios a_{ij} are zero, then the input mesh is interpolated.*

Proof Let P_i be the ith original mesh point neighbor of P. Then, regardless of the value of α, β and γ,

$$\begin{aligned} C_i = P' = C_i' = \tilde{b}_{1,i} = \tilde{b}_{2,i} = P, \\ b_{00,i} = b_{01,i} = b_{02,i} = b_{11,i} = b_{22,i} = P, \\ b_{03,i} = (7P + P_i)/8, \\ b_{04,i} = (P + P_i)/2, \end{aligned}$$

That is the boundary curve of the patch agrees with half an edge of the mesh. ■

In fact, it can be shown that all points except possibly $b_{12,i}$ lie on a bilinear surface generated by the four points that define the mesh region over which the patch is generated. Unfortunately any construction that allows complete flexibility in the sharpness of the blend, is polynomial and preserves symmetry will exit the convex hull of the input data at points surrounded by three patches.

Theorem 3.3. *No smooth, symmetry preserving, polynomial construction with adjustable blend can guarantee that the surface lies in the local convex hull of the input mesh where three patches join.*

Proof Abbreviate $u(t) := \lim_{a_{ij}\downarrow 0} D_2 p_i(0,t) = \lim_{a_{ij}\downarrow 0} D_1 p_{i+1}(t,0)$, $v(t) := \lim_{a_{ij}\downarrow 0} D_1 p_i(0,t)$, $w(t) := \lim_{a_{ij}\downarrow 0} D_2 p_{i+1}(t,0)$ where $\lim_{a_{ij}\downarrow 0}$ indicates that all blend ratios relevant to the local construction go uniformly to zero. Furthermore let u' denote the derivative of u with repect to t. If the input mesh is symmetric, say if the mesh outlines a cube, then tangent continuity and symmetry imply the polynomial equation

$$\lambda u = \mu(v + w) \qquad (*)$$

where, without loss of generality, λ and μ are scalar polynomials in t with $\lambda(0) = -1$ and $\mu(0) = 1$. Since we assume a single construction rule for all blend ratios, λ and μ should be analytic; otherwise we would have to distinguish cases. Hence the derivatives of λ and μ are finite for any choice of the blend ratios. If the blend can be adjusted to any degree of sharpness then the tangent plane has zero extent in the limit, i.e.

$$u(0) = 0 = (v + w)(0).$$

Differentiating both sides of Equation (*), we get $\lambda' u + \lambda u' = \mu(v' + w') + \mu'(v + w)$ and hence

$$-u'(0) = v'(0) + w'(0).$$

Since $u(0) = 0$ either $u'(0)$ or $-u'(0) = v'(0) + w'(0)$ point outside the convex hull unless $u'(0) = 0$. If $u'(0) = 0$, we can repeat the argument for the next higher derivative and ultimately get a contradiction since at least one of the derivatives of the boundary curve must be non-zero to span the distance from the mesh point to its neighbor. ■

Given the previous theorem, the construction proposed in Section 2 does as well as possible.

Proposition 3.4. *Let α, β and γ take on their default values and consider the surface in the neighborhood of a mesh point P with n neighbors. Then if either $n > 3$ and $a_{ij} \in [0,1]$ or $a_{ij} \in [10^{-6},1]$ then every point on the surface is a local convex combinations of the vertices of the input mesh points. If $n = 3$ and $a_{ij} \in [0,10^{-6}]$ then the violation of the convex hull is bounded by 10^{-6} times the furthest distance of P from one of its neighbors.*

Proof Since $0 \leq \beta \leq 1$,

$$C_i' = (1-\beta)P + \frac{1}{n}\sum_{k=1}^{n}(\beta + \alpha\cos(\frac{2\pi}{n}k))C_{i+k}$$

is a non negative combination of the C_{i+k} if $\alpha \leq \beta$. If $c := \cos(\frac{2\pi}{n}) > 0$, then inspection of $b_{11,i}$ and $b_{12,i}$ shows that all Bernstein-Bézier coefficients are convex combinations of the intermediate mesh points. Since the Bernstein-Bézier form evaluates by taking convex combinations of the coefficients, the surface lies in the convex hull.

If P has $n = 3$ neighbors, then $c = -1/2$ and, $b_{12,i}$ has an uncancelled negative component, $-C_{i,j+1}/32$. However, having one coefficient with a negative weight does not imply that the surface exits the convex hull. To obtain a bound on the possible violation of the convex hull predicted by the Theorem 3.3, we choose the coordinate system such that P is the origin. Since the worst case occurs when the negative weight in $b_{12,i}$ has maximal influence and the blend ratios at P are zero, we Taylor-expand p_i along the ray $v = ru$ in a neighborhood of the origin. Due to the zero blend ratios, the corresponding parametrization of the surface curve starts with a u^3 term.

$$u^3((\frac{1}{4}r - \frac{9}{32})P_i r^2 + (\frac{1}{4} - \frac{9}{32}r)P_{i+1}) + u^4((\frac{-3}{32}r + \frac{9}{16})P_i r^2 + (\frac{-3}{32} + \frac{9}{16}r)P_{i+1}r) + o(u^5).$$

On minimizing the factor multiplying P_i, we find the minimum value to be 10^{-7}, taken on for a small $u > 0$ and $r < 1/2$. In other words, the surface exits the local convex hull by at most 10^{-7} times the distance between P and its furthest neighboring original mesh point. Since increasing the blend ratios a_{ij} moves the patch into the convex hull by at least $a_{ij}\beta^*/6 = (7/48)a_{ij}$, the claim follows. ■

4. Remarks and observations

The main contribution of this paper is to give simple formulas for the Bernstein-Bézier coefficients of a C^1-surface spline based entirely on four-sided patches. The key to this construction is the particular choice of cubic boundary curves, and a use of degrees of freedom that yields a Bernstein-Bézier net as a refined mesh by averaging the original mesh points. (Not surprisingly, averaging results in more pleasing surfaces than curve interpolating algorithms like [Piper '87] or [Peters '90], and one may wonder what other than the tradition of Coons' construction attributes the curve mesh such preference when constructing a surface.) The analysis of the convex hull property, both the negative and the positive result, are of interest since they establish shape properties *a priori* as opposed to say the *a posteriori* display of curvature.

Comparing the biquartic splines with the cubic surface splines defined in [Peters '93], we find that the number of Bernstein-Bézier coefficients per subcell is 25 in either case. Moreover, both surface constructions can be joined to biquadratic patches where the mesh is regular. However, the biquadratic-cubic combination defined in [Peters '93] is superior in that it always satisfies the convex hull property. Also, combining the biquartic construction with biquadratic patches requires a transition layer of bicubic patches. A similar biquintic construction would be useful, if all cells of the input mesh are four-sided to start with, since then the mesh need not be refined.

Notes on the illustrations: The two cubes illustrate the maximal absolute curvature of a surface with $\alpha = 1/2$ uniformly whose mesh is intentionally chosen so that an interior square of the top face is interpolated. (To interpolate a whole face it suffices to set the blend ratios to zero.) Thus the dark area (blue in the original) is intentionally flat. The two sequences of figures illustrate the properties of the surface splines on larger data sets. The first sequence illustrates free-form modeling, built-in smoothness and the local convex hull property. The default overall blend ratio is 15% to keep the features focused. The second sequence shows how the shape parameters can be varied locally and globally. Thus the back face of the vice is interpolated (no violation of the convex hull property here, but intentional loss of tangent continuity) while the central cylinder of the base plate is rounded by setting the local blend ratios to 1/2 (the exact reproduction of cylindrical and other quadric features is subject of a forthcoming paper). From top to bottom, the default blend ratio is varied from 15% to 50% to 75%.

copy editor: I would be happy to send you TeX and postscript files.

References

W. Boehm, G. Farin, J. Kahmann, *A survey of curve and surface methods in CAGD*, Computer Aided Geometric Design1 (1984), 43–.

E. Catmull, J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, CAD 10, No 6 (1978): 350–355.

W. Dahmen, C.A. Micchelli, H.P. Seidel, *Blossoming begets B-splines bases built better by B-patches*, Mathematics of Computation, Vol. 59, No. 199, July 1992: 97–115.

D. Doo, *A subdivision algorithm for smoothing down irregularly shaped polyhedrons*, Proceedings on interactive techniques in computer aided design, Bologna (1978): 157–165.

G. Farin, *Curves and surfaces for computer aided geometric design*, Academic Press, 1990.

T.N.T. Goodman, *Closed surfaces defined from biquadratic splines*, Constructive Approximation 7 1991, 149–160.

J.A. Gregory, *Smooth parametric surfaces and n-sided patches*, Computation of curves and Surfaces, W. Dahmen, M. Gasca and C.A. Micchelli, eds., Kluwer Academic Publishers, Dordrecht, 1990: 457–498.

J. M. Hahn, *Filling polygonal holes with rectangular patches*, Theory and Practice of geometric modeling, W. Straßer and H.-P. Seidel eds., Springer 1989.

K. Höllig, H. Mögerle, *G-splines*, Computer Aided Geometric Design 7 (1989): 197–207.

S.L. Lee, A. A. Majid, *Closed smooth piecewise bicubic surfaces*, ACM Transactions on Graphics 10,4 (1991): 342–365.

C. Loop, T. DeRose, *Generalized B-spline surfaces of arbitrary topology*, Computer Graphics 24,4(1990): 347–356.

J. Peters, *Smooth mesh interpolation with cubic patches*, CAD **22**(2), 1990: 109–120.

J. Peters, *Smooth splines over irregular meshes built from few polynomial pieces of low degree*, CSD-TR-93-019, March 1993; to appear as C^1*-surface splines* in SIAM J. of Numerical Analysis 32-2.

J. Peters, *Smoothing vertex-degree bounded polyhedra*, to appear in ACM TOG.

A.R.M. Piah, *Construction of smooth surfaces by piecewise tensor product polynomials*, CS Report 91/04 University of Dundee, UK, 1991.

B. Piper, *Visually smooth interpolation with triangular Bézier patches* In: Geometric Modeling (G. Farin ed.), 1987.

U. Reif *Biquadratic G-spline surfaces*, Preprint 93-4, Math Inst A, Universität Stuttgart, 1993, to appear in CAGD.

R.F. Sarraga, G^1 *Interpolation of Generally Unrestricted Cubic Bézier curves*, Computer Aided Geometric Design 4(1-2):23–40,1987, R. Sarraga. Errata: G^1 interpolation of generally unrestricted cubic Bézier curves, CAGD **6**, 2, 1989, pp 167–172.

H.-P. Seidel, *Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles*, Constr. Approx. 7(1991): 257–279.

J.J. van Wijk, *Bicubic patches for approximating non-rectangular control-point meshes*, Computer Aided Geometric Design 3, No 1 (1984): 1–13.