SIAM J. Numer. Anal.
Vol. 1, No. 1, pp. 000–000, October 1993

C^1 SURFACE SPLINES *

JÖRG PETERS †

Abstract. The construction of quadratic C^1 surfaces from B-spline control points is generalized to a wider class of control meshes capable of outlining arbitrary free-form surfaces in space. Irregular meshes with non quadrilateral cells and more or fewer than four cells meeting at a point are allowed so that arbitrary free-form surfaces with or without boundary can be modeled 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, evaluates by averaging and obeys the convex hull property. For a regular region of the input mesh, the representation reduces to the standard quadratic spline. In general, any surface spline can be represented by Bernstein-Bézier patches of degree two and three. According to the user's choice, these patches can be polynomial or rational, three-sided, four-sided or a combination thereof.

Key words. C^1 surface, corner cutting, box splines, blending, geometric continuity, spline mesh, free-form surface modeling.

AMS subject classifications. 65D17,10,07 65Y25 68U07,05,

1. Introduction.

Splines assembled from B-splines are widely used to represent functions and surfaces. They combine a low degree polynomial or rational representation of maximal smoothness with a geometrically intuitive variation of the surface in terms of the coefficients: by connecting the coefficients one obtains a mesh that roughly outlines the surface. Repeated refinement of this mesh by knot insertion results in a sequence of meshes whose points are averages of the preceding and whose limit is the surface itself. In addition to an elegant algebraic definition this yields an alternative geometric characterization of the splines useful for establishing many shape properties of spline surfaces. But the tensor product B-spline representation has a major shortcoming. It cannot model certain real world objects without singularity in the parametrization, because, like a fishing net, each point in the interior of the B-spline mesh must be *regular*, that is surrounded by exactly four quadrilateral mesh cells. This makes it impossible to choose for example the boundary mesh of a cube as input and in fact restricts the topological type of the objects that can be modeled by the splines. Even if the object is a deformation of the plane, say a cube attached to a plane, and can thus in principle be outlined by a regular mesh, it may be more natural to join three or five quadrilateral mesh cells at the top and base points of the cube or to use non quadrilateral cells to model the feature. Using trimmed (non uniform rational B-) splines does not solve this problem since the trimming destroys one of the chief advantages of the B-spline representation, its built-in smoothness so that one ends up with the tricky task of smoothly joining the trimmed pieces. The goal is therefore to devise a representation that removes the regularity restrictions from the input mesh and yields a unified approach to surface modeling. The approach should reduce to the B-spline paradigm wherever the mesh is regular and have the following additional properties.

*Received by the editors April 15, 1993; accepted by the editors October 15, 1993. This work was supported by NSF grant CCR-9396164.

†Department of Computer Science, Purdue University, W-Lafayette IN 47907-1398 (jorg@cs.purdue.edu).

- *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, surface splines form a vector space of geometrically smooth surface parametrizations. In order to manipulate surface splines, it suffices to add, subtract 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.* Interpolation of input mesh points and normals can be done 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 a sum of the mesh points with coefficients that are positive and sum to one.
- *intuitive shape parameters.* The averaging process is geometrically intuitive. Parameters analoguous 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 summary, one would like a representation that departs as little as possible from the widely used non uniform rational B-spline standard and combines the intuitive cutting paradigm (see e.g. [4]) with a low degree parametrization.

The representation described in this paper generalizes the quadratic C^1 spline paradigm for both tensor-product B-splines and four-direction box splines: the input mesh serves as control mesh and blend or cut ratios act like knot spacings. The resulting surface is parametrized by Bernstein-Bézier patches that are, according to the user's choice, polynomial or rational, three-sided, four-sided or a combination thereof. (For details on the Bernstein-Bézier form and box spline form see [3], [12] and [5]. The three-sided and mixed surface spline developed in this paper satisfy *all* the above properties. If the user chooses to model using only four-sided patches, then, at vertices with $2n > 4$ neighbors, smoothness is traded for simplicity and low degree and in general only the residual of the smoothness constraints is minimized at the vertex. For surfaces built from three-sided patches or a combination of three-sided and four-sided patches, the surface is guaranteed to be tangent plane continuous everywhere.

Related work. The present method defines a spline space over irregular meshes by local averaging as do the algorithms in [23], [24] and [20]. The first of the three earlier algorithms contributes the idea of mesh refinement to separate irregular vertices, the second adds the quadratic spline representation and the third gives a recipe for obtainig a quadratic curve mesh. The main improvements of the present over the earlier algorithms are as follows.

- Fewer surface pieces are generated. While the earlier schemes generate 16 four-sided or 48 three-sided Bernstein-Bézier patches for each regular mesh point, the current algorithm can do with one fourth that number.
- The surface can be guaranteed to lie in the convex hull of the control mesh for the full range of blend ratios (knot spacings).
- Parametrization with three-sided and four-sided patches is treated uniformly and can be mixed.

[28], [13] and [17] pioneered the idea of geometrically continuous spline spaces (see also [25], [26]). [28] considers only special meshes, while the later constructions are more general but less localized in that one needs to solve large linear, irregularly sparse systems of equations to match data. This makes it more difficult to reason a priori about the shape of the resulting surface. Algorithms for generalized subdivision ([27], [10], [6], [18], [11], etc.) have been popular with the computer graphics community due to their simplicity and intuitive corner cutting interpretation. Unfortunately, these averaging schemes do not provide an explicit parametrization of the resulting surface at irregular mesh points. This not only makes it tricky to establish elementary properties like tangent plane continuity of the limit surface, but is also a major obstacle for integrating the techniques with other numerical and computer aided design representations. The work on reparametrization and geometric smoothness by numerous researchers (see [14] for a survey) is an essential tool for deriving surface splines. Algorithms like [29], [31], [16] have constructed smooth surfaces from meshes of data, the latter even with kth order continuity. These approaches differ from the present in that local constraint systems have to be solved to enforce patch to patch smoothness which make it difficult to predict the shape of the resulting surface; surface splines, in contrast, have explicit formulas and built-in continuity. Free-form surface splines based entirely on four-sided patches generalize the construction of [32, Section 3.5] to the case where $n > 4$ patches meet at a mesh point; the representation based on three-sided patches can be viewed as a special case of [21] using a well-chosen mesh of quadratic boundary curves as input. A-patches [1] (see also [15],[8]), B-patches ([30], [7]) and S-patches [19] provide elegant new solutions to the smoothing problem at the cost of a currently non standard patch representation.

Overview. The remainder of the paper consists of the definition and the proof of properties of the first-order free-form surface splines. Section 2 defines the splines in terms of their Bernstein-Bézier representation. Section 3 establishes the continuity and vector space properties of the splines. Section 4 establishes some shape properties of the resulting surfaces. Section 5 summarizes and the Appendix illustrates the representation.

2. An algorithm for refining an irregular mesh of points into a C^1 surface. Analogous to tensor product splines, a surface spline is defined by

- mesh points
- mesh connectivity
- blend ratios (knot spacings)

and an evaluation algorithm. Since the evaluation of polynomials in Bernstein-Bézier form is well known, it suffices to express the free-form surface spline in terms of the Bernstein-Bézier form. Below, this basis conversion is given a six step, geometrical interpretation. Steps 1-3 generate a refined mesh consisting of quadrilateral subcells and such that each original vertex is surrounded by vertices of degree four. Steps 4,5 and 6 generate the surface parametrization and depend on the user's choice of three-sided or four-sided or mixed patch representation.

1 Mesh refinement.
2 Edge cutting.
3 Quadratic meshing.
4 Quadratic patching.
5 Degree raising.
6 Twist adjustment.

The *input* to the algorithm is any mesh of points outlining a surface such that at most two mesh cells abut along any edge. The mesh may model a bivariate surfaces with or without boundary and of arbitrary topological genus (see [24] for a discussion of boundary conditions). Mesh cells need not be planar. Two additional properties of the input mesh are desirable and can be enforced by local preprocessing where necessary, e.g. by dividing offending cells (cf. Fig 5.4).

- *local degree-boundedness:* If V is a vertex of the cell c, and n and m are the number of neighbor of V and c respectively, then $\min\{n, m\} < 5$. Thus, if a cell has many edges, its vertices should have few neighbors.
- *projective convexity:* For each cell, there exists a projection of the cell vertices into a plane such that none of the projected vertices lies in the convex hull of the projected vertices.

Local degree-boundedness keeps the construction local and of least degree. Projective convexity is necessary to preserve the design intent, for example when an input cell is a facet of a boundary representation and has inner loops or holes. Such a cell has to be broken up to prevent the inner loops from being covered by the surface. For each vertex i of a cell, there are two scalar weights $0 < a_{ij} < 1$, $j = 1, 2$, called *blend ratios.* 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 (cf. Fig 5.3). The default is $a_{ij} := 1/2$. Algebraically, blend ratios play a role similar to relative knot spacings. Thus $a_{ij} = 0$ allows modeling sharp edges and to interpolate the mesh edges. The blend ratios of each cell may be modified independently from each other and other cells.

The *output* is a surface that follows the outline of the input mesh and consists, depending on the user's choice of surface representation, of either

(a) no more than $8e$ quadratic or cubic, three-sided patches that form a C^1 surface, where e is the number of edges of the input mesh; or
(b) no more than $2e$ biquadratic or bicubic, four-sided patches that form a C^1 surface except possibly at vertices and faces with $2n > 4$ neighbors (see Theorem 3.5); or
(c) a combination of biquadratic, four-sided patches covering regular mesh regions and cubic, three-sided patches for the remaining regions.

In the following, all coefficients V, C, P, etc. are points in space. All indices are interpreted modulo n.

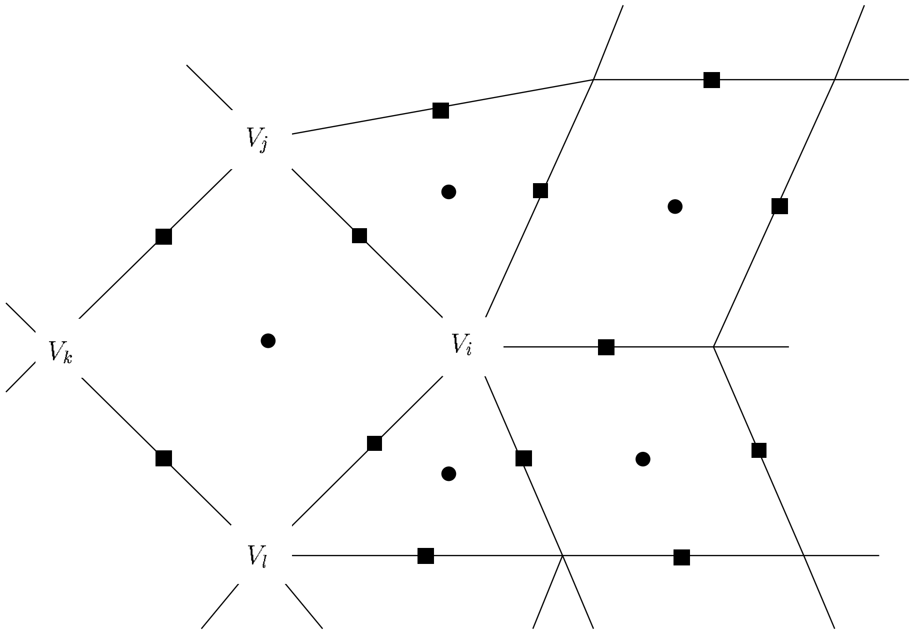
Step 1: Mesh refinement. For each edge with vertices V_i and V_j, add an edge vertex

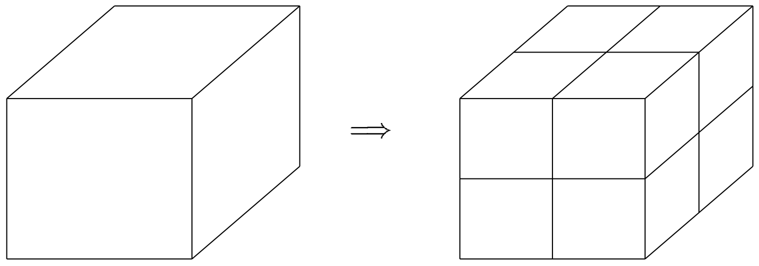
$$\blacksquare := (V_i + V_j)/2$$

and insert it between V_i and V_j. For each cell c with vertices $V_1, V_2, \ldots, V_m$, add a cell vertex

$$\bullet := \frac{1}{m}\sum V_i$$

and connect it to all edge vertices of the cell. This creates a refined mesh of quadrilateral subcells. (The construction remains valid if $\blacksquare$ and $\bullet$ are more general convex combinations of the vertices; here we state the default.)





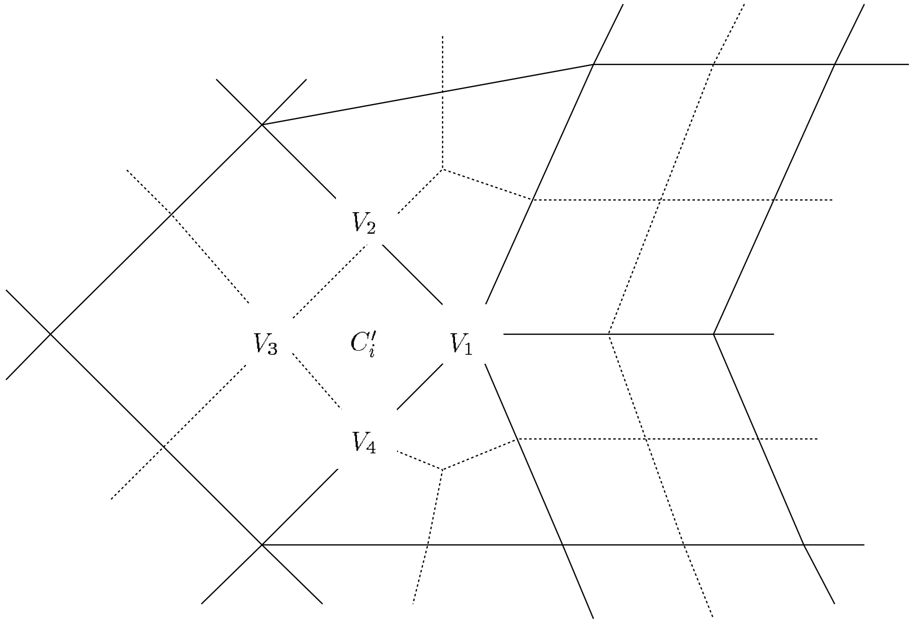
Step 2: Edge cutting. For each subcell c_i label the vertices V_1, V_2, V_3, V_4 in order, starting with the input mesh point, and define a preliminary subcell center C_i' subject to the blend ratios $0 < a_{i1}, a_{i2} < 1$ as the average

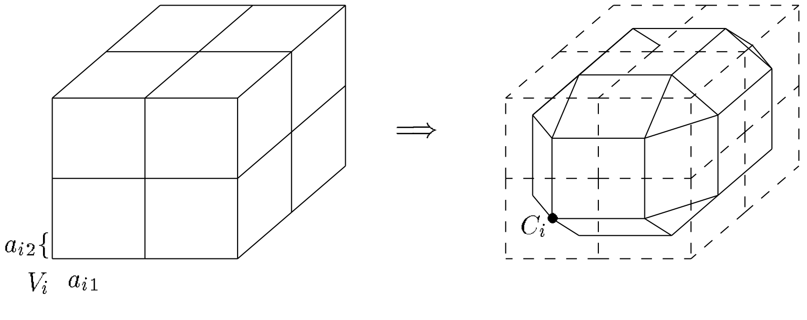
$$C_i' := (1-a_{i1})(1-a_{i2})V_1 + (1-a_{i1})a_{i2}V_2 + a_{i1}a_{i2}V_3 + (1-a_{i2})a_{i1}V_4.$$

Next derive subcell centers C_i from the C_i' such that all $(C_i + C_{i+1})/2$ surrounding a vertex lie in the same plane. For each vertex V surrounded by less than five cells, the subcell center is $C_i = C_i'$. For each vertex V surrounded in clockwise order by more than four cells $c_1, c_2, \ldots, c_n$, the cell center corresponding to c_i is

$$C_i := \bar{V} + \frac{2\omega_n}{n} \sum_j \cos(\frac{2\pi}{n} j) C_{i+j}'$$

where $\bar{V} := \frac{1}{n} \sum C_i'$, and $0 < \omega_n < 1$ with default $\omega_n^{-1} := \begin{cases} 1 + \cos\frac{2\pi}{n}, & \text{if } n \text{ is even} \\ 2\cos\frac{\pi}{n}, & \text{if } n \text{ is odd} \end{cases}$.
If V is to be interpolated, set $\bar{V} = V$ instead.



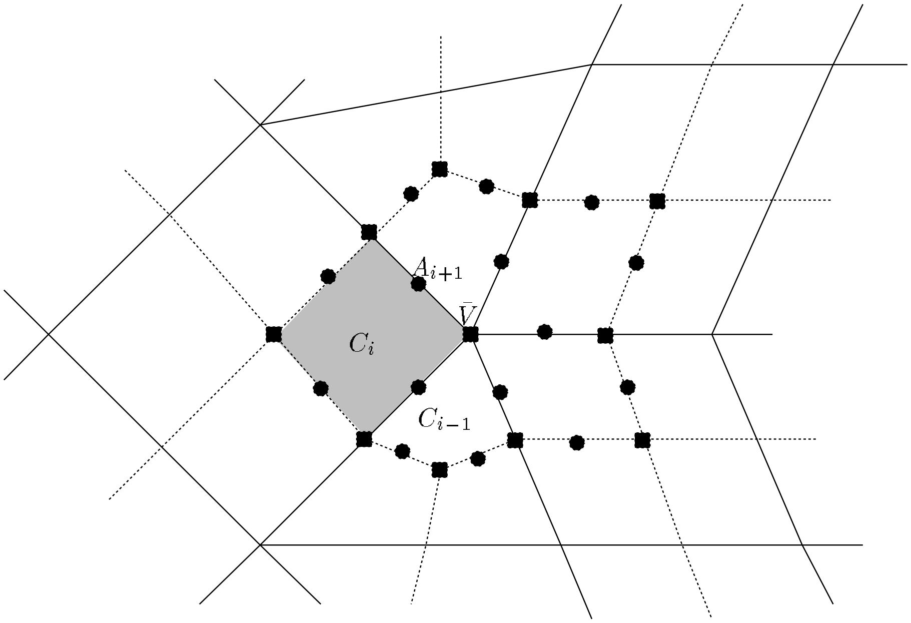


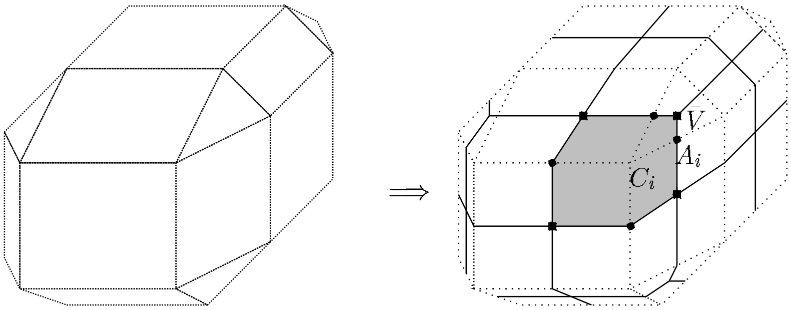
Step 3: Quadratic Meshing. For each vertex V surrounded in order by cell centers $C_1, C_2, \ldots, C_n$, compute the final location

$$\blacksquare := \bar{V} = \frac{1}{n}\sum C_i.$$

(Note that $\sum C_i = \sum C_i'$.) For each edge V, V_i separating two subcells with centers C_{i-1}, C_i, create an edge coefficient

$$\bullet \ := A_i := (C_i + C_{i-1})/2.$$





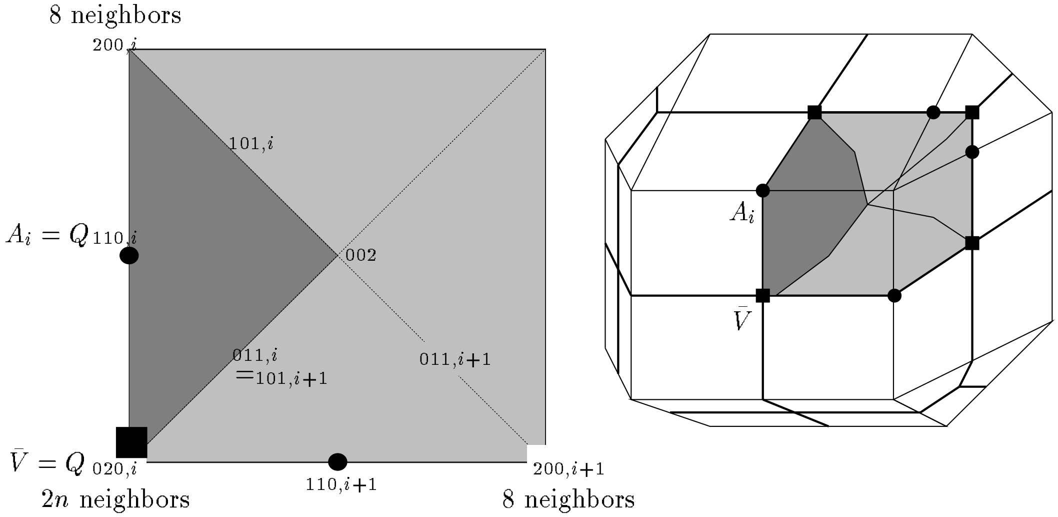
The output of Step 3 is a refined mesh such that three points are associated with each cell edge. These can be interpreted as the Bernstein-Bézier coefficients of a quadratic curve segment. Note that each coefficient $\blacksquare$ lies in the same plane as the surrounding A_i's.

The remaining three steps depend on the user's choice of three-sided or four-sided or mixed patch representation.

Surface splines based on three-sided patches.

Step 4T: Quadratic Patching. Each quadrilateral subcell is covered by four triangular patches as shown below. Thus regular mesh points now have eight neighbors. For each original vertex or cell vertex $\bar{V}$ set $Q_{020,i} := \bar{V}$, $Q_{110,i} := A_i$, $i = 1..n$ and compute the interior coefficients by averaging:

$$Q_{011,i} = Q_{101,i+1} := \frac{Q_{110,i} + Q_{110,i+1}}{2}, \qquad Q_{002} := \frac{Q_{101,i} + Q_{011,i+1}}{2}.$$



Step 5T: Degree raising. The quadratic patches are reinterpreted as cubic patches using the following formulas (*ed coefficients are recomputed in Step 6T)

$$\begin{array}{cccccccccc}
P_{300} & & & & & Q_{200} & & & \\
& P_{201} & & & & & \frac{Q_{200}+2Q_{101}}{3} & & \\
P_{210} & & P^*_{102} & & & \frac{Q_{200}+2Q_{110}}{3} & & \frac{Q_{002}+2Q_{101}}{3} & \\
& P^*_{111} & & P^*_{003} & = & & \frac{Q_{110}+Q_{101}+Q_{011}}{3} & & Q_{002} \\
P_{120} & & P^*_{012} & & & \frac{Q_{020}+2Q_{110}}{3} & & \frac{Q_{002}+2Q_{011}}{3} & \\
& P_{021} & & & & & \frac{Q_{020}+2Q_{011}}{3} & & \\
P_{030} & & & & & Q_{020} & & &
\end{array}$$

Step 6T: Twist Adjustment. At a vertex $\bar{V} = P_{030} = Q_{020}$ surrounded by n quadrilateral cells, set $c := \cos(\frac{2\pi}{n})$ and compute with C_i the center of the ith subcell

$$P_{111,i} = [(2-c)Q_{110,i} + (1+c)Q_{200,i} + 2C_i + Q_{020,i}]/6$$

Finally, counting with j around the four-neighbor vertex Q_{002}, average

$$P_{102,j} = P_{012,j-1} := (P_{111,j} + P_{111,j+1})/2, \qquad P_{003,j} := (P_{102,0} + P_{102,2})/2.$$

Steps 5T and 6T are not necessary if $n = 4$.

Surface splines based on four-sided patches.

Step 4R: Quadratic Patching. Associated with each subcell are four vertex, four edge and one center coefficient that may be interpreted as the Bernstein-Bézier coefficients of a biquadratic patch. For example, four of the coefficients are $Q_{00} = \bar{V}$, $Q_{10} = A_i$, $Q_{01} = A_{i+1}$ and $Q_{11} = C_i$.

Step 5R: Degree raising. For each subcell, the associated biquadratic patch is rewritten as a bicubic patch. That is,

$$\begin{bmatrix} Q_{00} & Q_{10} & Q_{20} \\ Q_{01} & Q_{11} & Q_{21} \\ Q_{02} & Q_{12} & Q_{22} \end{bmatrix} \Longrightarrow \begin{bmatrix} P_{00} & P_{10} & P_{20} & P_{30} \\ P_{01} & P_{11}^* & P_{21} & P_{31} \\ P_{02} & P_{12} & P_{22}^* & P_{32} \\ P_{03} & P_{13} & P_{23} & P_{33} \end{bmatrix} =$$

$$= \begin{bmatrix} Q_{00} & \frac{2Q_{10}+Q_{00}}{3} & \frac{2Q_{10}+Q_{20}}{3} & Q_{20} \\ \frac{2Q_{01}+Q_{00}}{3} & \frac{4Q_{11}+2(Q_{01}+Q_{10})+Q_{00}}{9} & \frac{4Q_{11}+2(Q_{21}+Q_{10})+Q_{20}}{9} & \frac{2Q_{21}+Q_{20}}{3} \\ \frac{2Q_{01}+Q_{02}}{3} & \frac{4Q_{11}+2(Q_{01}+Q_{12})+Q_{02}}{9} & \frac{4Q_{11}+2(Q_{21}+Q_{12})+Q_{22}}{9} & \frac{2Q_{21}+Q_{22}}{3} \\ Q_{02} & \frac{2Q_{12}+Q_{02}}{3} & \frac{2Q_{12}+Q_{22}}{3} & Q_{22} \end{bmatrix}$$

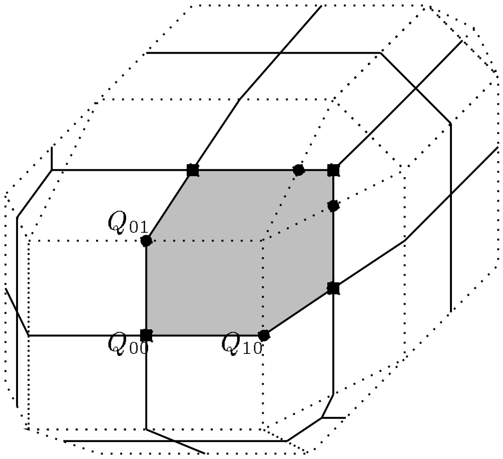
Step 6R: Twist Adjustment. For each vertex V surrounded clockwise by patches p_i, $i = 1..n$, let $V = P_{00,i}$ and $P_{01,i-1} = P_{10,i}$ and define

$$E_i := P_{10,i} + \frac{c}{3}(P_{30,i} - P_{20,i}), \quad c := \cos(\frac{2\pi}{n}).$$

Then we recompute

$$P_{11,i-1} = \begin{cases} -\sum_{j=1}^{n}(-1)^j E_{i+j} & \text{if } n \text{ is odd} \\ \frac{-2}{n}\sum_{j=1}^{n}(n-j)(-1)^j E_{i+j} & \text{if } n \text{ is even.} \end{cases} \tag{2.1}$$

Steps 5R and 6R are not necessary if $n = 4$.

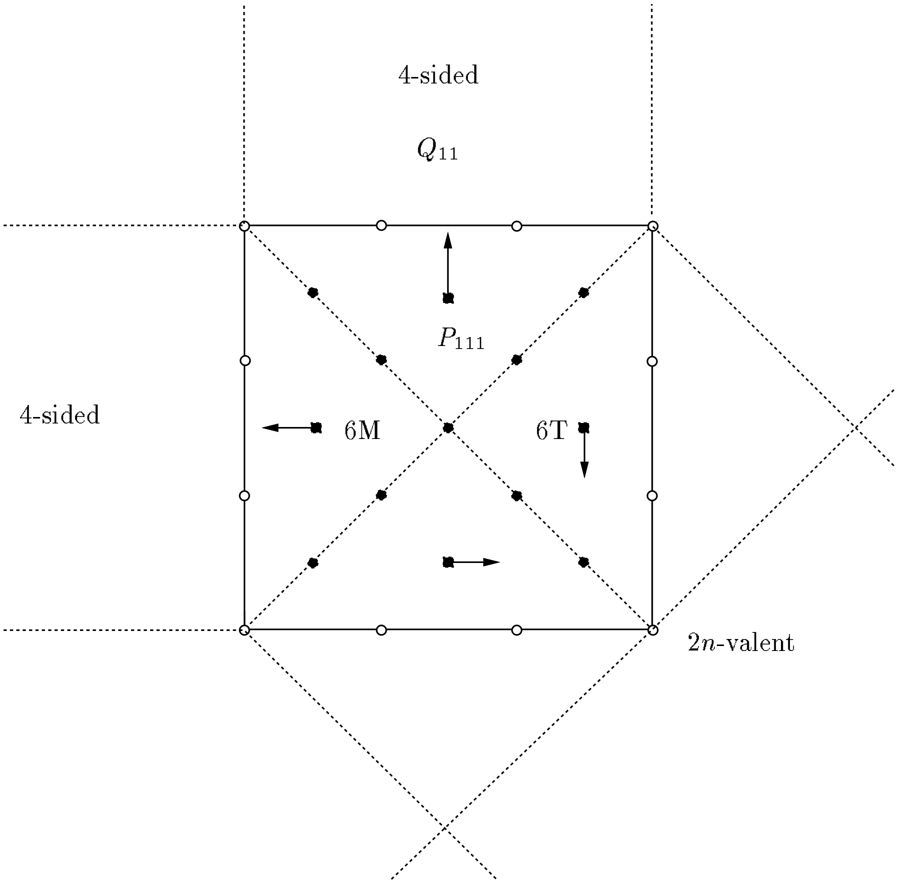


Surface splines based on three-sided cubics joining biquadratic patches. To join a three-sided patch p smoothly to a biquadratic four-sided patch q (that corresponds to a regular part of the mesh; see Remark 3.8 for the more general case), denote as Q_{00}, Q_{10}, and Q_{20} the coefficients of the common quadratic boundary and as Q_{11} the center coefficient of the biquadratic patch. Patch p is constructed by steps $4T$ and $5T$ as usual but the perturbation step determines the center coefficient as follows.

Step 6M: Twist Adjustment.

$$P_{111} = Q_{10} + \frac{1}{3}(\frac{Q_{20}+Q_{00}}{2} - Q_{11}).$$

The join allows filling n-sided holes in a biquadratic tensor-product spline surface with three-sided cubic patches. A hole is first divided into n quadrilateral cells. Then each cell is covered by four three-sided, cubic patches. Since the center coefficients of the cubic patches attached to the $2n$-valent central vertex are not edge-adjacent to the biquadratic patches, they need only be adjusted as in Step 6T.



3. Continuity and vector space properties. The previous section defined the surface spline subject to the input mesh and the blend ratios. This section shows that splines based on the same mesh connectivity and ratios form a vector space of tangent continuous maps. (Oriented) tangent plane continuity is characterized as the agreement of the derivatives of two maps p and q from $\mathbb{R}^2$ to $\mathbb{R}^n$ after reparametrization by a map φ from $\mathbb{R}^2$ to $\mathbb{R}^2$ that connects the domains Ω_p and Ω_q of p and q:

$$p = q \circ \varphi \quad \text{and} \quad D_1 p = D_1(q \circ \varphi) \quad \text{along } E_p$$

where $\varphi(E_p) = E_q$, E_p and E_q are edges of Ω_p and Ω_q respectively. D_1 denotes differentiation in the direction perpendicular to E_p and φ maps interior points of Ω_q to exterior points of Ω_p thus avoiding cusps. We prove oriented tangent plane continuity first for the construction with three-sided patches, then for the construction with four-sided patches and finally for the mixed construction. We prepare the result with two lemmas. The first records the mesh structure after the refinement step.

LEMMA 3.1. *Steps 1–3 generate a mesh of quadrilaterals bounded by quadratic curves and such that at least one vertex of every edge has exactly 4 neighbors.*

LEMMA 3.2. *The coefficients A_i constructed in Step 3 at V satisfy*

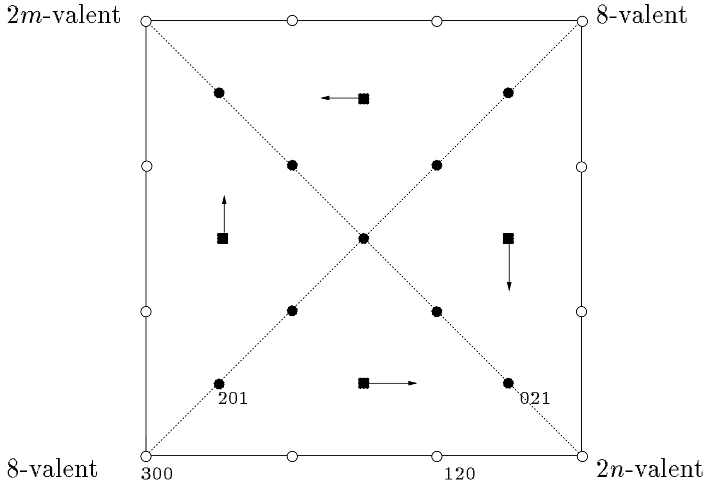
$$A_{i-1} - 2cA_i + A_{i+1} = 2(1-c)V, \quad i = 1..n, \quad c := \cos(\frac{2\pi}{n}).$$

Proof. For simplicity we center the coordinate system at $V = \frac{1}{n}\sum C_i = 0$. For $n = 3$ and $n = 4$, the statement follows directly from $A_i := \frac{C_{i-1}+C_i}{2}$. For $n > 4$, Step 2 enforces

$$\begin{aligned} &C_{i-1} - 2cC_i + C_{i+1} \\ &= \frac{2\omega_n}{n}\sum_{i=1}^{n} C_i\big(\cos(\frac{2\pi}{n}(i-1)) + \cos(\frac{2\pi}{n}(i+1)) - 2\cos(\frac{2\pi}{n})\cos(\frac{2\pi}{n}i)\big) = 0 \end{aligned}$$

and hence the same holds for the averages A_i. □

By Lemma 3.1 and Step 4T, each three-sided patch has one vertex with eight neighbors and one vertex with four neighbors. Therefore, the central coefficient P_{111} is associated with at most one vertex that is neither 8-valent nor 4-valent. The arrows in the diagram below indicate this association. (○ labels the degree-raised boundaries generated by Step 5T, ● the averaged coefficients of the interior boundaries and ■, represents the central coefficient P_{111} constructed in Step 6T.)



Theorem 3.3. *Steps 1–3 and 4T–6T of the algorithm generate a quadratic-cubic C^1 surface.*

Proof. Step 6T enforces along E_p

$$D_1 p_{j-1} = D_1(p_j \circ \psi), \text{ where } \psi(t_1,t_2) := \mathrm{id}(t_1,t_2) + t_1 \begin{bmatrix} 0 \\ 1-ct_2 \end{bmatrix} \quad \text{and } c := \cos(\frac{2\pi}{n})$$

for adjacent patches p_i and p_{i-1}. Let Q_{200}, Q_{110} and Q_{020} be the Bernstein-Bézier coefficients of the common quadratic boundary curve; Q_{200} corresponds to $t_2 = 0$ where 4 quadrilaterals (or 8 triangles) meet and Q_{020} corresponds to $t_2 = 1$ where n quadrilaterals meet. Then the constraints on the coefficients of the cubic patches are

$$\begin{aligned}
P_{201,j-1} - P_{300,j} + P_{201,j} - P_{300,j} &= \frac{2}{3}(Q_{110} - Q_{200}) \\
P_{111,j-1} - P_{210,j} + P_{111,j} - P_{210,j} &= \frac{1}{3}\big(Q_{020} - Q_{110} + (1-c)(Q_{110} - Q_{200})\big) \\
P_{021,j-1} - P_{120,j} + P_{021,j} - P_{120,j} &= \frac{2}{3}(1-c)(Q_{020} - Q_{110})
\end{aligned}$$

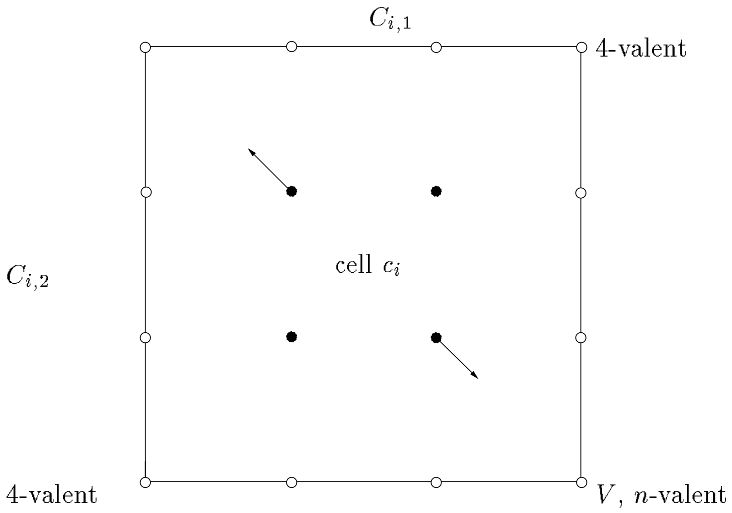
The first and the third equation hold by Lemma 3.2, the second is checked by substitution. □

Corollary 3.4. *Steps 4T-6T cover cells with no irregular mesh points with a quadratic four direction box spline.*

Proof. When all cell vertices are regular mesh points, then $c = 0$ in Step 6T and the construction leaves the patch unchanged as a quadratic after Step 4T. By Theorem 3.3, the surface is C^1 and the connecting map the identity. Therefore the patches join parametrically C^1 and are the quadratic four direction box spline corresponding to the subcell centers C_i (cf. [5]). □

Next, we analyse the four-sided patch construction of Steps 4R–6R. For $i = 1..n$, let c_i be the subcells that surround V and $C_{i,1}$ and $C_{i,2}$ the center coefficients of the two subcells that share an edge with c_i, but do not have V as a vertex. Define

$$E(V) := \frac{-1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{2} (-1)^{i+j} C_{i,j}.$$



THEOREM 3.5. *Steps 1–3 and 4R–6R of the algorithm generate a biquadratic-bicubic C^1 surface except at a vertex V with $2n > 4$ neighbors and $E(V) \neq 0$. In the latter case the residual of the C^1 constraints is minimized by the construction.*

Proof. Unless $m = 2n > 4$ and $E(V) \neq 0$, the construction enforces along the edge

$$D_1 p_{j-1} = D_1(p_j \circ \psi), \qquad \text{where } \psi(t_1,t_2) := \mathrm{id}(t_1,t_2) + t_1(1-t_2)^2 \begin{bmatrix} 0 \\ 2c \end{bmatrix}$$

and $c := \cos(2\pi/n)$. Let Q_{20}, Q_{11} and Q_{02} be the Bernstein-Bézier coefficients of the common quadratic boundary curve of p_{j-1} and p_j; Q_{20} corresponds to $t_2 = 0$ where 4 quadrilaterals meet and Q_{02} corresponds to $t_2 = 1$ where n quadrilaterals meet. The equation is verified by writing out the conditions in Bernstein-Bézier form:

$$\begin{aligned}
P_{30,i} - P_{31,i} &= P_{13,i-1} - P_{03,i-1} \\
P_{20,i} - P_{21,i} &= P_{12,i-1} - P_{00,i-1} \\
\frac{4c}{9}(Q_{20,i} - Q_{10,i}) + P_{10,i} - P_{11,i} &= P_{11,i-1} - P_{01,i-1} \\
\frac{4c}{3}(Q_{10,i} - Q_{00,i}) + P_{00,i} - P_{01,i} &= P_{10,i-1} - P_{00,i-1}
\end{aligned}$$

The first, second and fourth constraint are satisfied by the choice of ■ in Step 2. With $P_{30,i} - P_{20,i} = \frac{2}{3}(Q_{20,i} - Q_{10,i})$, and $P_{10,i} = P_{01,i-1}$, the third constraint is

$$\frac{P_{11,i} + P_{11,i-1}}{2} = E_i := P_{10,i} + \frac{c}{3}(P_{30,i} - P_{20,i}).$$

Choosing $P_{11,i}$ according to (2.1) enforces the constraint if n is odd:

$$P_{11,i-1} + P_{11,i} = -\sum_{j=1}^{n}(-1)^j(E_{i+j-1} + E_{i+j}) = 2E_i.$$

If n is even, then

$$\begin{aligned}
P_{11,i-1} + P_{11,i} &= \frac{-2}{n}\sum_{j=1}^{n}(n-j)(-1)^j(E_{i+j-1} + E_{i+j}) = \frac{-2}{n}\Big(\sum_{j=1}^{n}(-1)^j E_j - nE_i\Big) \\
&= 2E_i - 2\Big(\sum_{j=1}^{n}(-1)^j E_j\Big)/n.
\end{aligned}$$

That is, the total residual $2\sum_{j=1}^{n}(-1)^j E_j$, a multiple of $E(V)$, is equally distributed over the patch boundaries emanating from the vertex. □

COROLLARY 3.6. *Steps 4R-6R cover cells with no irregular mesh points with a biquadratic (B-)spline.*

Proof. If both vertices of an edge e have valency 4, then the transversal derivatives of the patches p_i and p_{i-1} that share the edge are collinear and of equal size. Hence the patches are joined parametrically C^1 and the construction agrees with the construction of biquadratic splines from the control points C_i. □

THEOREM 3.7. *Steps 1–3 and 4T,5T and 6M of the algorithm generate a biquadratic-cubic C^1 surface.*

Proof. Due to Theorems 3.3 and 3.5, we need only consider the transition between q, a biquadratic patch and p a cubic patch. Then along E_p

$$D_1 p = D_1(q \circ \psi), \text{ where } \psi(t_1, t_2) := \mathrm{id}(t_1, t_2) - \frac{t_1}{2}\begin{bmatrix}1\\1\end{bmatrix}.$$

Let Q_{00}, Q_{11} and Q_{20} be the Bernstein-Bézier coefficients of the common quadratic boundary curve; $Q_{00} = P_{300}$ corresponds to $t_2 = 0$ and Q_{20} corresponds to $t_2 = 1$. Then the constraints on the Bernstein-Bézier coefficients are

$$\begin{aligned} Q_{10} - Q_{00} &= Q_{01} - Q_{00} + 3(P_{201} - P_{300}),\\ \frac{Q_{20} - Q_{00}}{2} &= Q_{11} - Q_{10} + 3(P_{111} - P_{210}),\\ Q_{20} - Q_{10} &= Q_{21} - Q_{20} + 3(P_{021} - P_{120}). \end{aligned}$$

The first and the third constraint are satisfied by the quadratic construction. Since $P_{210} = \frac{Q_{00}+2Q_{10}}{3}$, the second constraint is

$$P_{111} = \frac{1}{3}(Q_{00} + 2Q_{10} + Q_{10} - Q_{11} + \frac{Q_{20} - Q_{00}}{2}) = Q_{10} + \frac{1}{3}(\frac{Q_{20} + Q_{00}}{2} - Q_{11}).$$

□

Remark 3.8. A biquadratic patch can be joined to a three-sided patch at an n-sided vertex using the reparametrization

$$\psi := \mathrm{id} + t_1 \begin{bmatrix} -\frac{1}{2} + t_2(\frac{1}{2} - 2c) \\ -\frac{1}{2} \end{bmatrix}$$

and

$$P_{111} = Q_{10} + \frac{1}{3}(\frac{Q_{20} + Q_{00}}{2} - Q_{11}) - \frac{c}{3}(Q_{10} - Q_{00}).$$

Theorem 3.9. *C^1 surfaces generated from input meshes with the same connectivity, choice of three-sided and four-sided patches, and blend ratio for each subcell form a vector space.*

Proof. Two surfaces generated from input meshes with the same connectivity have a natural correspondence of patches. Consider two smoothly abutting patches p_i and q_i, $i = 1, 2$ of the ith surface. The connecting map depends only on the connectivity of the mesh via c. Identifying the open neighborhood of the edges $E_{p,1}$ and $E_{p,2}$ as E_p, there is a single connecting map φ such that

$$p_i = (q_i \circ \varphi) \quad \text{and} \quad D_1 p_i = D_1(q_i \circ \varphi) \quad \text{along } E_p.$$

Consequently, if $p_0 := \beta_1 p_1 + \beta_2 p_2$ and $q_0 := \beta_1 q_1 + \beta_2 q_2$, then

$$p_0 = (q_0 \circ \varphi) \quad \text{and} \quad D_1 p_0 = D_1(\beta_1 p_1 + \beta_2 p_2) = \beta_1 q_1 \circ \varphi + \beta_2 q_2 \circ \varphi = D_1(q_0 \circ \varphi)$$

along E_p as claimed. □

As pointed out in the next section, coalescing knots results in a reduction of the smoothness of the spline in the associated directions. Breaks can be modeled by meshes with boundary. Spreading the xy-coordinates in the plane and choosing all z coordinates of the mesh equal to zero except for one, generates a hump familiar from B-splines. By the convex hull property, Proposition 4.2, these basis functions form a nonnegative partition of unity.

4. Shape properties of the resulting surface. This section establishes the convex hull property of surface splines. Additionally it is shown that surfaces are flat in the neighborhood of a mesh point if and only if the mesh is locally flat and that the edges of the input mesh are interpolated and thus the outlines of the input polytope recaptured when the blend ratios are zero. Additionally it is shown that the centroid of an input mesh cell is interpolated, if the blend ratios of the cell are all equal.

The following lemma contains the essence of the proof of the convex hull property.

LEMMA 4.1. *After Step 2 the cell centers C_i lie in the same plane and $A_i := \frac{C_i+C_{i-1}}{2}$ lie in the convex hull of the preliminary subcell centers points C'_i.*

Proof. Choosing the local coordinate system such that $\bar{V} := \frac{1}{n}\sum_j C'_j = 0$, we want to find $C_i \neq 0$ in the null space of the constraint matrix corresponding to the smoothness constraints $C_{i-1} - 2\cos(\frac{2\pi}{n})C_i + C_{i+1} = 0$, $i = 1..n$ (indices modulo n). This null space is spanned by the cyclical permutations $v_k := \big(\cos\frac{2\pi}{n}(\ell+k)\big)_{\ell=1..n}$, $k = 1..n-1$ of the vector $v_0 := (\cos\frac{2\pi}{n}\ell)_{\ell=1..n}$. Weighting each v_k by $\omega_n C'_k$ gives a uniform solution

$$C_i := \frac{2\omega_n}{n}\sum_j \cos(\frac{2\pi}{n}j)C'_{i+j}.$$

(The normalization by $\frac{2}{n}$ leaves the vertices of a regular n-gon with x-coordinates $\cos(\frac{2\pi}{n}j)$ invariant if $\omega_n = 1$ since $\sum_j \cos^2(\frac{2\pi}{n}j) = \frac{1}{2}\sum_j 1 + \frac{1}{2}\sum_j \cos 2\frac{2\pi}{n} = \frac{n}{2}$.) The coplanarity of the C_i follows from Lemma 3.2. Define

$$a_{i,j} := \frac{1+\omega_n\cos(\frac{2\pi}{n}(j-i)) + \omega_n\cos(\frac{2\pi}{n}(j-i+1))}{n}.$$

Then $a_{i,j} \geq 0$, $\sum_{j=1}^n a_{i,j} = 1$, $i = 1..n$, $n \geq 3$, and $A_i = \sum_j a_{i,j}C'_j$. □

The contraction by $\omega_n \in (\frac{1}{2}, 1]$ is also necessary due to the following extreme example. If $C'_j := (-1,0,0)$ for $j = 1..n-1$ and $C'_n := (n-1,0,0)$, then the x-component of C_i is

$$\frac{2\omega_n}{n}\Big(-\sum_{l=1}^{n}\cos\big(\frac{2\pi}{n}(i+l)\big) + n\cos\big(\frac{2\pi}{n}(n+i)\big)\Big) = 2\omega_n\cos\frac{2\pi}{n}i$$

and the constraint that the A_i lie in the convex hull is $\omega_n(-\cos(\frac{2\pi}{n}(i+1)) - \cos(\frac{2\pi}{n}i)) < 1$ for all i. If $\omega_n > 0$, the left hand side is maximized for $i = \lfloor\frac{n}{2}\rfloor$. If n is even, then the maximum is $1+\cos\frac{2\pi}{n}$, if n is odd, then the maximum is $2\cos\frac{\pi}{n}$.

Since all coefficients are computed as convex averages of the C_i and the C_i themselves are convex combinations of the input mesh points, the following proposition holds.

PROPOSITION 4.2. *If the construction uses three-sided patches at irregular vertices, i.e. Steps 4T, 5T, 6T or 6M, then the surface is in the convex hull of the input mesh.*

Example 4.3. The following example shows, that Step 6R does not always yield planar surfaces for planar cells and therefore does not stay within the convex hull of the input mesh. Consider the corner $(0,0,0)$ of a cube with neighbor vertices $V_1 = (1,0,0)$, $V_2 = (0,1,0)$, and $V_3 = (0,0,1)$. Then the twist coefficient of the bicubic

$$P_{11,1} = E_2 - E_3 + E_1 = \frac{1}{9}\begin{bmatrix}-1\\1\\-1\end{bmatrix}$$

is outside the cube. This problem can be fixed using higher degree or more patches, but at this point I prefer to keep the construction simple and the degree low.

PROPOSITION 4.4. *The curvature at V is zero if and only if the neighboring vertices of the refined mesh, $V_i, i = 1..n$ lie in the same plane as V.*

Proof. Let $P(n)$ be the component of P normal to the tangent plane at V. If $V_i(n) \neq 0$ for some i, then the curvature of the ith boundary curve is nonzero. Conversely, if all V_i lie in the tangent plane, then all boundary curves are coplanar. In the case of four-sided patches this implies $E_i(n) = 0$ and hence $P_{11,i}(n) = 0$. In the case of three-sided patches this implies $P_{111,i}(n) = \frac{1+c}{6} V_i(n) = 0$. □

Zero cut ratios result in a singular parametrization at the input mesh point. Just as for B-splines with coalesced knots, the degree of smoothness drops by one. This has also a desirable consequence in that the C^1 surface degenerates into a C^0 surface that tightly interpolates the input mesh.

PROPOSITION 4.5. *An edge between two cells with zero transversal cut ratios is interpolated. Planar cells with zero cut ratios are covered by a planar surface if represented with three-sided patches.*

Proof. Zero cut ratios coalesce all cell centers C_i surrounding an original mesh point V into V. The boundary curves are convex averages of the C_i, hence coincide with the edges. Up to Step 5 (T,R or M), all coefficients are convex combinations of points on the edges and planarity follows. For a three-sided or mixed construction

$$\begin{aligned} &Q_{020,i} = Q_{110,i} = Q_{011,i} = Q_{110,i+1} = V \qquad && Q_{200,i} = Q_{110,i-1} = V_i \\ &Q_{101,i} = \frac{Q_{200,i} + Q_{020,i}}{2} = \frac{V_i + V}{2} && P_{111,i} = \frac{(5-c)V + (1+c)V_i}{6} \end{aligned}$$

proving planarity. □

To model with the vector space it is good to know that the resulting surfaces can interpolate certain averages of the input data even if the interpolation is not forced in Step 2 by setting $\bar{V} = V$.

PROPOSITION 4.6. *Let $a_{i,j}$, $i = 1..n$, $j = 1,2$ be the blend ratios and C_i the centers of the n subcells surrounding a cell centroid V. If $a_{i,j} = a_j$ and $C_i = C'_i$, then the surface generated by the algorithm interpolates the cell centroid.*

Proof. Let V_i be the vertices of the cell. Step 3 determines the vertex of the refined mesh corresponding to the cell as

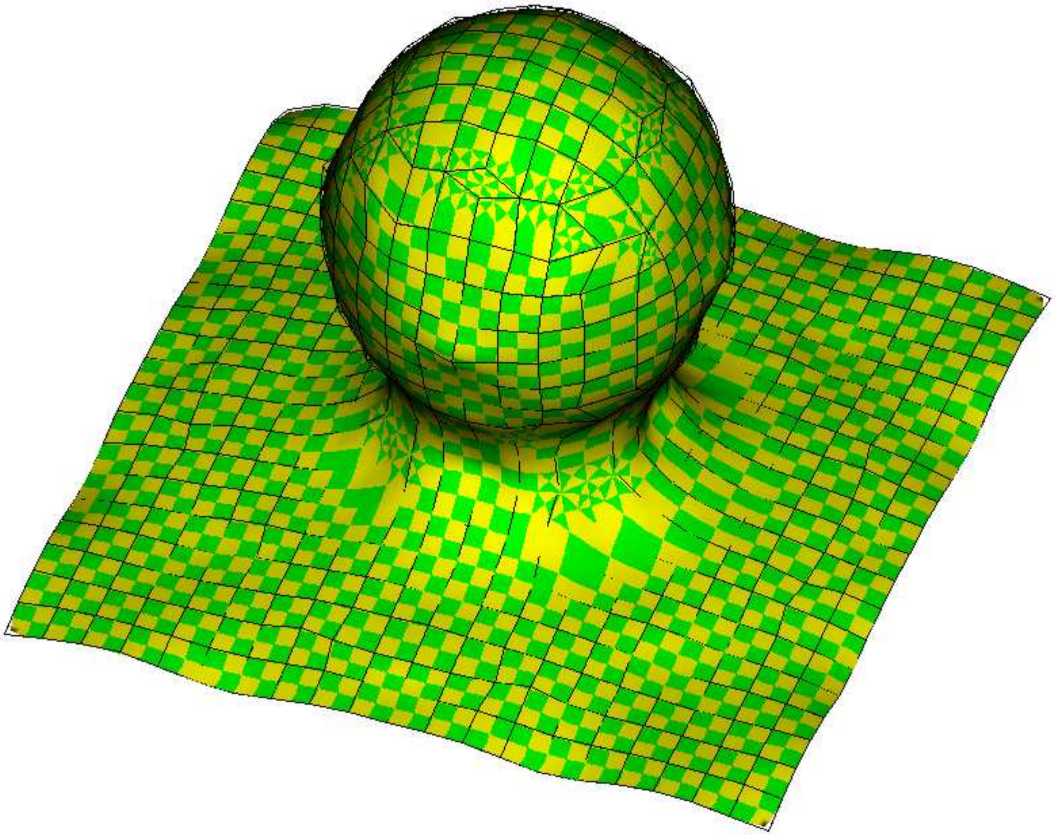
$$\begin{aligned} \bar{V} &= \frac{1}{n}\sum C_i \\ &= \frac{1}{n}\sum[(1-a_{i1})(1-a_{i2})V_i + (1-a_{i1})\frac{a_{i2}}{2} + (1-a_{i-1,1})\frac{a_{i-1,2}}{2} \\ &\quad + (1-a_{i2})\frac{a_{i1}}{2} + (1-a_{i+1,2})\frac{a_{i+1,1}}{2} + \frac{1}{n}\sum_l a_{l,1}a_{l,2}]V_i \end{aligned}$$

In the particular case, the bracketed expression sums to one. Since $\bar{V}$ is a vertex coefficient of the patches meeting at $\bar{V}$ it is interpolated. □

Thus a uniform choice of blend ratios suffices to interpolate for example the centroids of a cube while remaining within the convex hull of the input mesh.

5. Examples. The following three examples illustrate uses and properties of surface splines.

Fig. 5.1



The surface of Figure 5.1 combines four-sided biquadratic patches and three-sided cubic patches to model a geological formation courtesy of M. Peters of Sierra Geophysics. The input mesh consists of 664 points and 693 cells. On a generic Silicon Graphics Indigo workstation the combined computation and display time is on the order of a few seconds.

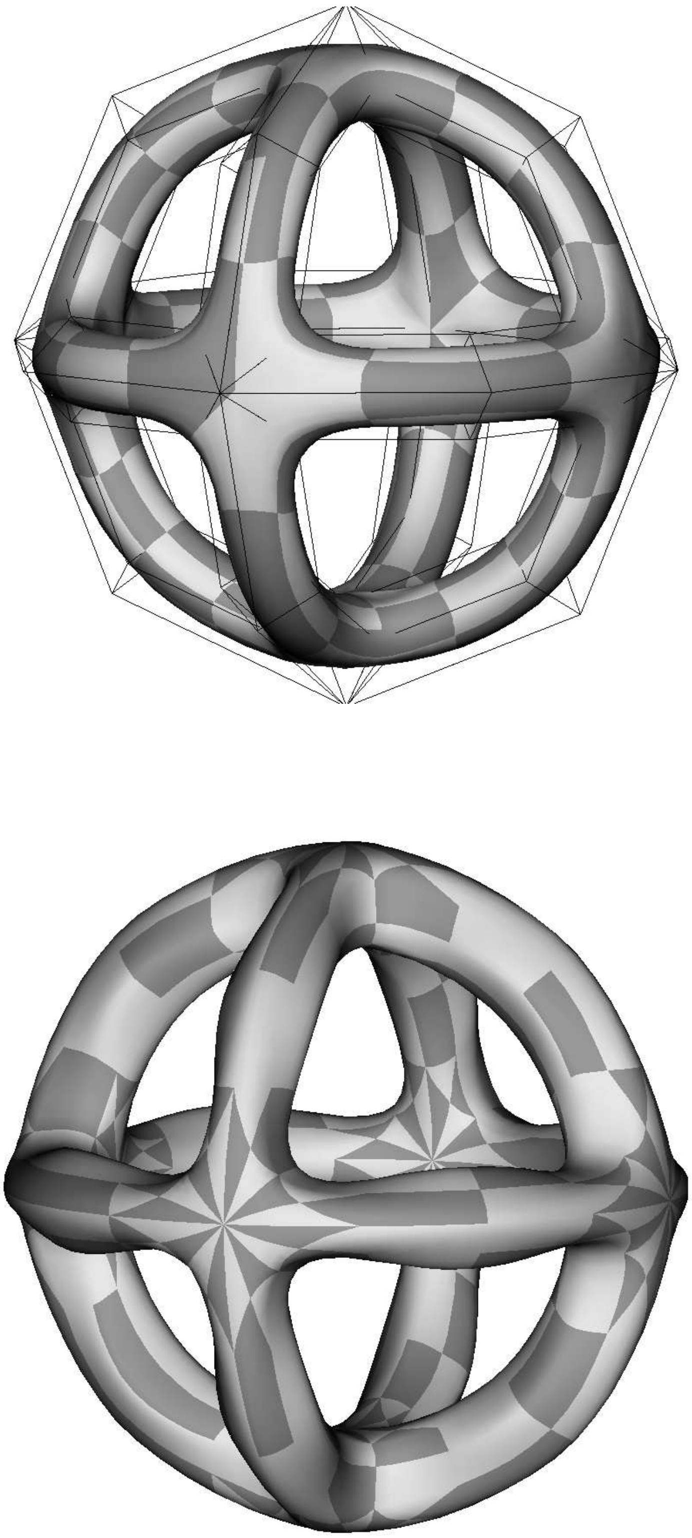


Fig. 5.2

Fig. 5.3

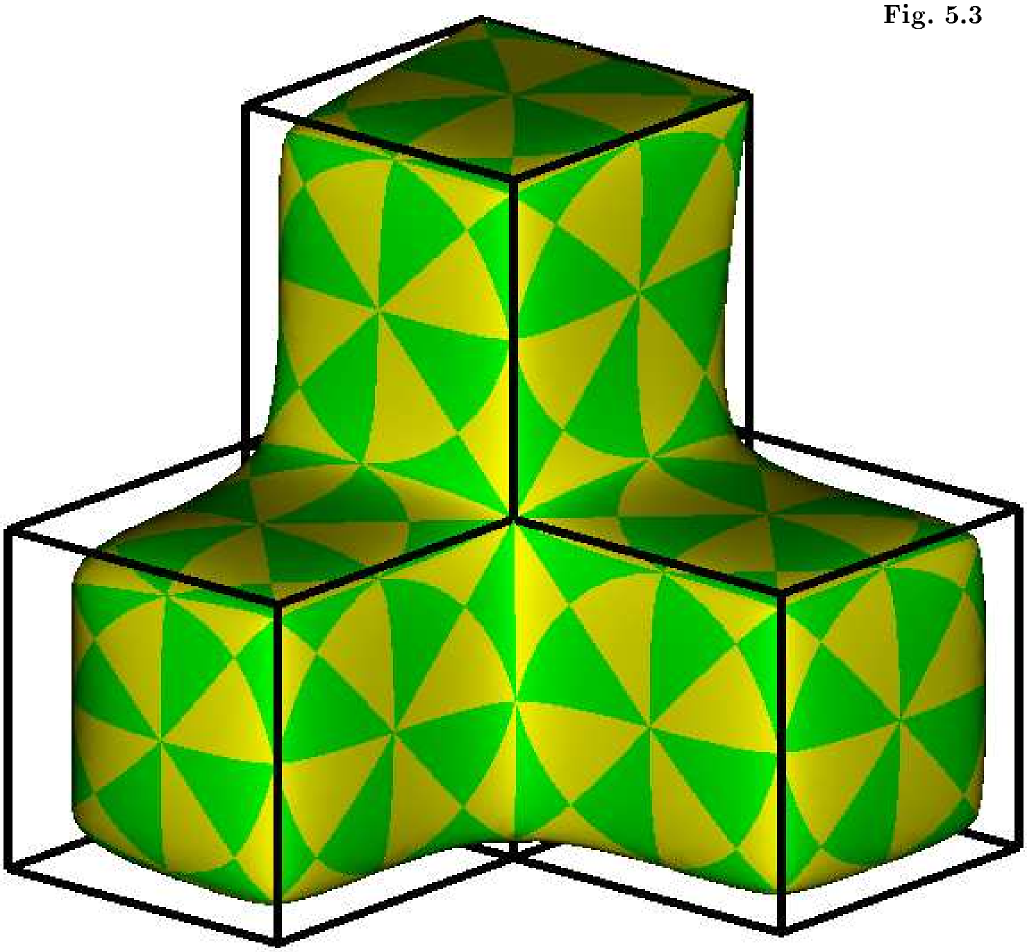
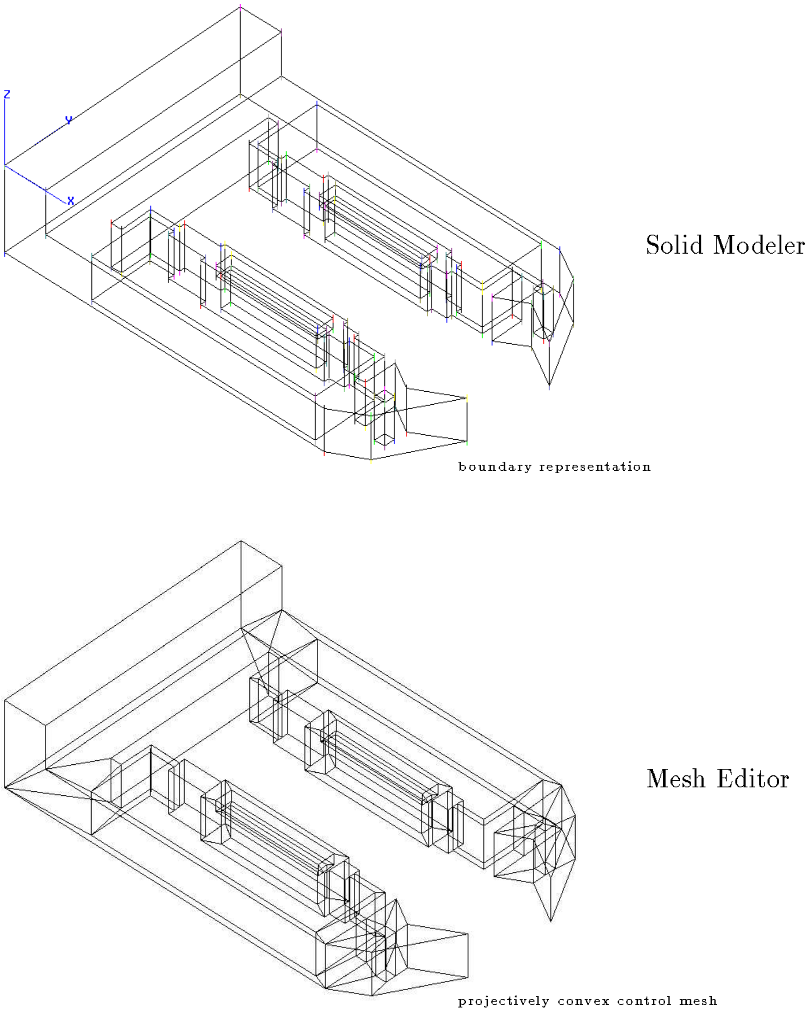


Figure 5.2 shows two surfaces of higher topological genus. The mesh is a boundary representation generated by a solid modeler. In the top display, the mesh is approximated using exclusively four-sided patches. In the bottom display, the mesh is interpolated using the local interpolation provided in Step 2 of the algorithm. For variety, a mixed representation is chosen for the bottom surface. The mesh could have just as well been interpolated with four-sided patches only.

Figure 5.3, a multicube constructed from three-sided patches, illustrates selective blending. The overall blend ratio is 0.1. However, at the upper right edge the ratio is set to 0.0 to obtain a sharp edge that interpolates the input edge and on the left edge of the left cube the ratio is 0.25 so that the suitcase corners are more rounded than the rest of the object.

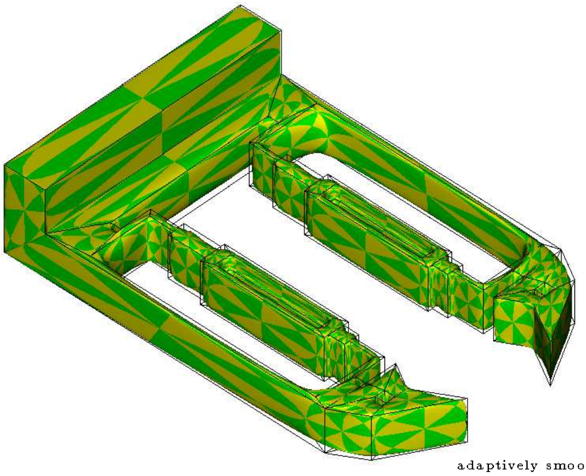
Figure 5.4 illustrates preprocessing of a mesh to enforce projective convexity and illustrates selective blending to obtain both sharp and rounded edges. The original mesh is courtesy of the Computer-Aided Design and Graphics Laboratory, Engineering Research Center for Intelligent Manufacturing Systems, Purdue University. The mesh editor was written by M. Ivory and M. Tripunitara based on code by M. Mantyläa.

Fig. 5.4



6. Conclusion. The algorithm in Section 2 defines a surface representation that generalizes quadratic C^1 splines. It smoothes a general, regular or irregular mesh of points into a C^1 surface parametrized according to the user's choice by quadratic and cubic three-sided patches, biquadratic and bicubic four-sided patches, or biquadratic and cubic patches. In the case of a purely four-sided representation some continuity and shape properties are traded for low degree and simplicity of construction. Also, the convex hull property does not hold for all blend ratios. Where the mesh is regular, the surface is quadratic. Input meshes with the same connectivity and the same blend ratio for corresponding cells give rise to a vector space of surface splines. This and the convex hull property are useful for approximating and locally editing the spline surface. The role of the knot spacing is played by geometrically intuitive blend ratios.

Fig. 5.4 ctd.



Surface Splines

adaptively smoothed object

Zero blend ratios result in a C^0 surface that tightly interpolates the input mesh. It is possible to interpolate the input mesh without solving a global sparse system of equations analoguous to interpolation by a quadratic spline at every second knot.

Due to the built-in smoothness, the representation reduces the number of unknowns for such problems as shape improvement of smooth surfaces and related differential equations on surfaces. Integrals and derivatives can be determined for the Bernstein-Bézier representation whose coefficients in turn are defined in terms of the control mesh and the blend ratios. Free-form surface splines can be used to smooth and blend the boundary representation of a solid model.

REFERENCES

[1] C. Bajaj and J. Chen and G. Xu, *Interactive Modelling with A-Patches*, Computer Science Technical Report, CAPO-93-02, Purdue University, 1993.

[2] W. Boehm, *Generating the Bézier points of triangular splines*, R.E. Barnhill, W. Boehm (eds.), North Holland, 1983, 77–91.

[3] W. Boehm, G. Farin, J. Kahmann, *A survey of curve and surface methods in CAGD*, Computer Aided Geometric Design 1 (1984), 1–60.

[4] C. W. de Boor, *Local corner cutting and the smoothness of the limiting curve*, Computer Aided Geometric Design , 7; 1990: 389–397;

[5] C. W. de Boor, K. Höllig, S. Riemenschneider, *Box splines*, Springer Verlag, NY, 1994.

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

[7] 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.

[8] W. Dahmen, T.-M. Thamm-Schaar, *Cubicoids: modeling and visualization*, Computer Aided Geometric Design , Vol. 10, 1993, 93-108.

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

[11] N. Dyn, D. Levin, D. Liu, *Interpolatory convexity preserving subdivision schemes for curves and surfaces*, preprint, 1992.

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

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

[14] 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.

[15] B. Guo, *Modeling arbitrary smooth objects with algebraic surfaces*, PhD thesis, Computer Sciences, Cornell University, 1991.

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

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

[18] C. T. Loop, *Smooth subdivision surfaces based on triangles*, Master's thesis, University of Utah, 1987.

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

[20] C. T. Loop, *Smooth low degree polynomial spline surfaces over irregular meshes*, preprint, 1993.

[21] J. Peters, *Smooth Mesh Interpolation with Cubic Patches*, CAD, 22 No 2 (1990), 109–120.

[22] J. Peters, *Smooth Interpolation of a Mesh of Curves*, Constructive Approximation, 7 (1991), 221–247.

[23] J. Peters *Constructing C^1 surfaces of arbitrary topology using biquadratic and bicubic splines*, May 1992, to appear in Designing fair curves and surfaces, N. Sapidis (ed.).

[24] J. Peters, *Smooth free-form surfaces over irregular meshes generalizing quadratic splines*, Computer Aided Geometric Design, 10 (1993) 347–361.

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

[26] U. Reif *Biquadratic G-spline surfaces*, Preprint 93-4, Math Inst A, Universität Stuttgart,1993.

[27] M. Sabin, *The use of piecewise forms for the numerical representation of shape*, PhD thesis, Hungarian Academy of Sciences, Budapest, Hungary, 1976.

[28] M. Sabin, *Non-rectangular surface patches suitable for inclusion in a B-spline surface*, in P. ten Hagen (ed.), Proceedings of Eurographics '83, North Holland, 57–69.

[29] 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.

[30] 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.

[31] L. Shirman, C. Séquin, *Local surface interpolation with Bézier patches* Computer Aided Geometric Design 4, No 4, 279–296.

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