UNIVERSITY OF WISCONSIN-MADISON
CENTER FOR THE MATHEMATICAL SCIENCES

Fitting smooth parametric surfaces to 3D data

by
Jörg Peters

Technical Summary Report # 91-2
August 1990

Abstract

If Computer Aided Geometric Design is to deliver on its promise to rationalize and speed up the process of designing physical objects, it must master the construction of free-form surfaces. A key challenge is to control smoothness while matching given data and keeping the surface representation simple. This thesis both derives and applies new techniques to meet the competing requirements.

In particular, the thesis develops several equivalent notions of first- and second-order smoothness between patches, based on geometric invariants and of higher-order smoothness, based on the chain rule. Further, it exhibits the necessary and sufficient constraints on the data that allow enclosing a vertex by a C^1-complex of patches, or, for the symmetric case, by a C^k-complex of patches, and derives four techniques to overcome the enclosure problem. By classifying techniques for smoothly connecting patches and for enclosing vertices, a large number of algorithms in the literature are characterized. Variations of five new, implemented algorithms are described, each representing a different approach to supplying levels of smoothness for different types of data. For example, a method employing triangular cubic patches is shown to interpolate a mesh of curves with regular oriented tangent plane continuity and a surface assembled from tensor product patches of degree $2k + 2$ yields k^{th}-order smoothness while matching data of the same order along patch boundaries.

AMS (MOS) Subject Classifications: 41A15, 41A10, 41A05

Key words: Bernstein-Bézier form, geometric continuity, interpolation, vertex enclosure, computer aided geometric design, free-form surface subdivision

---

This research was supported by NSF DMS-8701275

FITTING SMOOTH PARAMETRIC SURFACES TO 3D DATA

Jörg Peters

Under the supervision of Professor Carl de Boor
at the Department of Computer Sciences
University of Wisconsin-Madison

Abstract If Computer Aided Geometric Design is to deliver on its promise to rationalize and speed up the process of designing physical objects, it must master the construction of free-form surfaces. A key challenge is to control smoothness while matching given data and keeping the surface representation simple. This thesis both derives and applies new techniques to meet the competing requirements.

In particular, the thesis develops several equivalent notions of first- and second-order smoothness between patches, based on geometric invariants and of higher-order smoothness, based on the chain rule. Further, it exhibits the necessary and sufficient constraints on the data that allow enclosing a vertex by a C^1-complex of patches, or, for the symmetric case, by a C^k-complex of patches, and derives four techniques to overcome the enclosure problem. By classifying techniques for smoothly connecting patches and for enclosing vertices, a large number of algorithms in the literature are characterized. Variations of five new, implemented algorithms are described, each representing a different approach to supplying levels of smoothness for different types of data. For example, a method employing triangular cubic patches is shown to interpolate a mesh of curves with regular oriented tangent

plane continuity and a surface assembled from tensor product patches of degree $2k+2$ yields k^{th}-order smoothness while matching data of the same order along patch boundaries.

To my parents

Acknowledgements

Writing a thesis is not always enjoyable and it takes a good adviser and friends to keep up the motivation. Carl de Boor sets an example through his work. I appreciate that he helped out with the formulation of results and prompted me to look more closely at underlying structures. His research grant financed my assistantship. Meera Sitharam, Srinivasan, Kirk Haller, Sze-ping Wong, and Amos Ron were my most frequent victims for sketching new ideas. Klaus Höllig's summer project started my research. I benefited from talking to many researchers in my field and thank, in particular, Ray Sarraga for his continued interest.

Table of Contents

Chapter One

Introduction

If Computer Aided Geometric Design is to deliver on its promise to rationalize and speed up the process of designing physical objects, it must master the construction of free-form surfaces; that is, of surfaces that do not necessarily arise as the graph of a function over the plane, but that are allowed to be closed and bent to suit the application. A key challenge for practical use is to control smoothness while matching given data and, at the same time, keeping the surface representation simple. Smoothness of certain parts is not just an aesthetic consideration but often dictated by functionality, e.g. to improve fluid flow. Generalizing from functions over a fixed, planar domain to free-form surfaces or manifolds entails a natural generalization of the notion of smoothness between surface pieces (patches) to a nonlinear constraint whose special linearized version agrees with familiar notions of derivative continuity for bivariate functions. Underlying is the fact that a surface need only locally be represented (parametrized) by a function, and that this parametrization changes as the surface is traversed. Smoothness competes with interpolation requirements in a subtle fashion that becomes apparent when many patches abut and enclose a vertex. For example, not every mesh of curves meeting at a vertex can be interpolated by a regular smooth patch complex. Smoothness' clash with simplicity becomes apparent when smoothing a geodesic dome to a sphere: each patch has to be able to flex and is no longer a simple, linear image of a flat domain.

This thesis both derives and applies new techniques to meet the competing requirements: smoothness, simplicity, confinement of features (localness) and interpolation. In particular, the thesis derives several equivalent notions of first- and second-order smoothness between patches based on geometric invariants and of higher-order smoothness based on the chain rule. It champions the notion of *regular oriented tangent plane continuity* for first-order continuity by showing that existence and uniqueness of a normal along a common boundary curve between two patches is equivalent to the matching of the first-order transversal derivatives after suitable reparametrization. Special attention is given to proper orientation of the tangent plane, to avoid cusps and other singularities, and to the 'weight' functions that arise from the reparametrization. Since the first and second fundamental form exhaust the geometric invariants in three dimensions, and since continuity can be equally characterized by applying reparametrization and the chain rule, the analysis of higher-order continuity relies on this characterization. Connecting-maps, i.e. smooth, invertible maps that relate the domains of two patches, play a major role in this, since they allow tracing a directional derivative from one patch across the common boundary into the neighboring patch. The thesis shows how the topology of the data restricts the choice of connecting-maps.

Enclosing a common vertex by several smoothly connected patches is a nontrivial task: not every mesh of curves with a welldefined tangent plane at the mesh points has a smooth regularly parametrized piecewise polynomial interpolant with one patch per mesh facet. For first-order smoothness, this is the result of propagation of second-order information across C^2 patches compounded with rank deficiency of the corresponding constraint matrix at vertices with an even number of neighbors (see Chapter 3). A detailed analysis yields four techniques to overcome the problem: forcing the curve mesh to meet second-order data, using two or more patches between each pair of curve segments (splitting), using rational patches with singularities at the vertex (Gregory's technique), or parametrizing curve segments singularly at the vertex. From this and the analysis of the vertex enclosure problem for a natural bilinear connecting-map and k^{th}-order smoothness (Chapter 6), it becomes apparent that the 2-direction (rectangular) mesh is a fortuitous singularity among all topological configurations: by separating constraints, it allows for a solution of lower degree than can be expected in general. This explains both the elegance of the construction and the difficulty in extending methods like [Gordon '69] to arbitrary topologies.

Rather than overwhelming the reader with the literature that has accumulated over the last three decades, this thesis offers a classification of some 20 algorithms for the construction of smooth, interpolating surfaces with piecewise polynomials: Chapter 4 characterizes the methods by their combination of techniques for connecting two patches and resolving the vertex enclosure problem. Also underrepresented in this thesis are the authors' efforts in implementing a test bed for algorithms that takes care of such tedious tasks as assembly of patches of arbitrary degree and in an arbitrary topology, evaluation, rendering and interrogation. The test bed is available to students of the area.

Despite the general analysis, the ultimate goal of this thesis is the generation of algorithms that are sufficiently robust and flexible and are optimal with respect to smoothness, simplicity and localness so that they can be included into a design package. Each of the five algorithms presented in Section 5 shows a different approach to supplying levels of smoothness for different types of data. For example, a robust method employing triangular cubic patches is shown to interpolate a mesh of curves with regular oriented

tangent plane continuity, and a surface assembled from tensor-product patches of degree $2k+2$ yields k^{th}-order smoothness while matching data of the same order along patch boundaries.

1.2 Notation

The input data to the surface construction algorithms is a *mesh of data*, i.e., consists of two components: information on connectedness (combinatorial structure) and numerical data (geometric detail). A mesh of data consists minimally of points in 3-space, also called *data point* or *mesh point* or *vertex*, and (logical) links between them. A *j-point* is a data point with j neighbors and an *even-point* is a data point with an even number of neighbors. A *mesh curve* is a curve (segment) between two data points; it consists generically of just one (polynomial) piece. To each minimal circuit (hole) in the mesh of logical links corresponds a *facet* of the mesh. A facet can be covered by one or more patches. A *patch* is a bivariate vector-valued map from the unit triangle or square to $\mathbb{R}^3$ that is k times continuously differentiable, where k depends on the application; in the following, patches are typically polynomial. Since the number of angles of the domain distinguishes total-degree from tensor-product patches, one customarily confuses domain and range and refers also to the patches as *triangular* and *rectangular*. Facets are bounded by *edges*, patches by *boundaries*, and surfaces by a *rim* (if any). Any boundary is the image of a straight line.

A polynomial patch of *total degree d* is represented in Bernstein-Bézier

-form (*BB-form*) as

$$x \mapsto \sum_{|\alpha|=d} \xi(x)^\alpha \binom{d}{\alpha} c(\alpha),$$

where $\alpha := (\alpha_1, \alpha_2, \alpha_3)$, $|\alpha| := \alpha_1 + \alpha_2 + \alpha_3$, $c(\alpha)$ are the (vector) coefficients, and the 3 linear polynomials (barycentric weight functions) ξ_i are defined by

$$\xi_1 p(0,0) + \xi_2 p(1,0) + \xi_3 p(0,1) = p \quad \text{for all } p \text{ with } \deg p \le 1.$$

Analogously, the bivariate *tensor-product* patch of degree (d,e) is defined by

$$(x_1, x_2) \mapsto \sum_{|\beta|=d} \sum_{|\gamma|=e} (\xi(x_1))^\beta (\xi(x_2))^\gamma \binom{d}{\beta}\binom{e}{\gamma} c(\beta,\gamma),$$

where $\beta := (\beta_1, \beta_2)$, $\gamma := (\gamma_1, \gamma_2)$, and ξ is defined by

$$\xi_1 p(0) + \xi_2 p(1) = p, \quad \text{for all } p \text{ with } \deg p \le 1. \tag{1.1}$$

The BB-form is the representation of choice, since this form gives the geometric meaning of 'control points' to its coefficients and easy access to value and derivative information along patch boundaries. Since [Farin '86] and [de Boor '87] already give a detailed account of the properties of the BB-form, only the salient points for the purpose of this thesis will be developed with the material.

It is good to keep in mind that parametric surfaces in 3-space are constructed by determining the *vector* coefficients of their polynomial pieces. Each piece maps from the unit square or unit triangle to 3-space. For the analysis of smoothness across boundaries, it will suffice to look at univariate polynomials, namely the derivatives of the patches along a boundary. Since

the main algebraic work consists of multiplying these univariate polynomials, it is advantageous to work with the modified BB-form

$$p : t \mapsto \sum_{j=0}^{d} t^j (1-t)^{d-j} b^j \tag{1.2}$$

and to write

$$p \sim [b^0, \ldots, b^d]$$

to indicate that p is given by (1.2). Thus

$$[b^0, \ldots, b^d][c^0, \ldots, c^d] = [b^0 c^0, \ldots, \sum_{k+l=j} b^k c^l, \ldots, b^d c^d],$$

where bc is the scalar product of b and c, and $p \sim [a^0, \ldots, \binom{d}{j} a^j, \ldots, a^d]$, in terms of its control points a^j. For example, raising the degree of the quadratic polynomial with control points a^0, a^1 and a^2 is expressed as $[1,1][a^0, 2a^1, a^2] = [a^0, 2a^1 + a^0, 2a^1 + a^2, a^2]$. If nothing else, the use of the modified BB-form avoids writing fractions and gives the k-fold *degree-raising* polynomial the simple form $[1,1]^k$. Greek letters are reserved for scalar polynomials. Superscripts number the coefficients and subscripts count the curves emanating from a given point. The counting is cyclic modulo the number of neighbors.

D_i denotes the partial derivative in the direction of the i^{th} unit vector $\mathbf{i}_i$. The (Fréchet-) derivative is denoted by D, i.e. $D := (D_1, D_2)$. We abbreviate

$$p_u := D_1 p_{|(u,0)} \quad p_v := D_2 p_{|(u,0)} \quad q_w := D_2 q_{|(w,0)},$$

and write

$$p^{ij} := D_1^i D_2^j p$$

for the i-fold derivative with respect to the first and the j-fold derivative with respect to the second argument of the bivariate map p. Note that p_u, p_v and q_w are univariate polynomials. To distinguish definitions from identities, ':=' and '=' are used judiciously, e.g. '=:' defines terms of the right hand side by those on the left.

1.3 Overview

Chapter 2 develops the notion of regular oriented tangent plane continuity by giving several equivalent characterizations. An example shows that the prevalent characterization of first-order smoothness in the literature is unnecessarily restrictive. The description of second-order smoothness is based on the second fundamental form.

Chapter 3 points out that fitting polynomial patches smoothly to a mesh of curves, as is customary in many applications, is, in general, not a well-posed problem regardless of the degree (and shape) of the patches used. Not every mesh that allows for a tangent plane at the mesh points has a smooth regularly parametrized interpolant with one patch per mesh facet. The problem arises when a vertex is enclosed by a k^{th}-order continuous complex of $k+1^{\text{st}}$-order continuous patches. Several techniques are shown to overcome the problem: use of rational patches with base points, of singular parametrizations, of split or macro patches, or of curves that match second-order data at the mesh points.

Chapter 4 classifies some algorithms in the literature that construct smooth, interpolating surfaces with polynomial pieces. The criteria are: generality of the interpolation conditions, flexibility in the number of edges a facet may have, degree of the polynomial pieces and type of reparametrization functions.

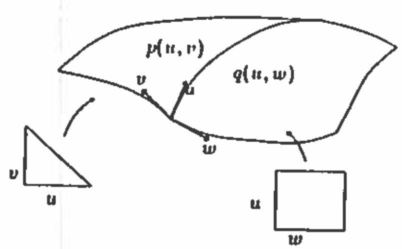
Chapter 5 gives a short description of the algorithms in [Peters '88a '88b/'90a '89b '89d] and describes improvements to remedy shortcomings of [Peters '88a '89d]. In particular, the algorithm in [Peters '88a] for C^1 surface interpolation with linearly varying boundary normal is improved by using *singular* quintic and biquartic patches in place of [bi]cubics in order to resolve the vertex enclosure problem. The algorithm in [Peters '89d] which creates 4-sided patches over a triangulation is improved by forcing the equiparametric 'diagonal' of the 4-sided patches to closely approximate a mesh of curves. The algorithm in [Peters '89b], which uses a single patch per facet, gives occasion to analyze the transition of an algorithm from general to special input data that allow for a simpler, lower degree interpolant.

Chapter 6 gives a definition of higher-order continuity based on the chain rule and shows how symmetries in the construction restrict the choice of the maps that connect the patch domains. For a bilinear connecting-map that reflects the change of topology from a 4-point to an n-point, a C^k interpolating construction with polynomials of degree $2k+2$ is developed.

Chapter Two

First-order continuity between two patches

This section derives the first-order continuity constraints enforced by the algorithms of Chapter 5. In particular, first-order continuity between parametric piecewise polynomial patches is characterized as ***regular oriented tangent plane continuity.*** Smoothness based on the second fundamental form is discussed at the end of the chapter and higher-order continuity in Chapter 6.



(2.1) Figure: Parametrization of abutting patches.

In the following, we consider two patches, p and q, abutting along a common boundary curve. The patches are differentiable; in particular, the definition of the univariate polynomials p_u, p_v and q_w in Section 1.2 makes sense.

(2.2) Lemma. [first-order continuity] *Consider two polynomial patches, p and q, that share a common boundary with parameter u. Then the following are equivalent.*

(i) The surface normal of p is welldefined and agrees with the normal of q at each point of the boundary:

$$\frac{p_v \times p_u}{\|p_v \times p_u\|} = \frac{p_u \times q_w}{\|p_u \times q_w\|}, \quad \text{[matching normal]} \qquad (2.3)$$

$$p_v \times p_u \neq 0. \quad \text{[non-vanishing normal]} \qquad (2.4)$$

(ii) The tangent planes are coplanar and the orientation agrees at each point of the boundary:

$$(p_v \times p_u)q_w = 0 \quad \text{[common tangent plane]} \qquad (2.e)$$

and

$$((p_v \times p_u) \times p_u)q_w > 0 \quad \text{[proper orientation]} \qquad (2.i)$$

(iii) There exist scalar-valued functions λ, μ and ν (of u) such that, at each point of the boundary,

$$\lambda p_u = \mu p_v + \nu q_w \quad \text{[common tangent plane]} \qquad (2.E)$$

$$p_v \times p_u \neq 0, \quad \mu\nu > 0. \quad \text{[proper orientation]} \qquad (2.I)$$

Proof. We will use the identities

$$(a \times b)(c \times d) = \det[(a \times b), c, d] = ((a \times b) \times c)d.$$

(i)⟹(ii): Multiply (2.3) by $q_w \|p_v \times p_u\|$ to see that (2.e) holds. Since $p_u(u)$ and $p_v(u)$ are linearly independent for any u, (2.3) implies that

$$0 < (p_v \times p_u)(p_u \times q_w) = ((p_v \times p_u) \times p_u)q_w.$$

(ii)$\Longrightarrow$(iii): Since $\det[p_v, p_u, q_w] = 0$, there exist scalar-valued functions λ, μ and ν such that (2.E) holds. Since $p_u(u)$ and $p_v(u)$ are linearly independent for any u, we may assume that $\nu > 0$ on the interval $[0,1]$. Substituting (2.E) into (2.i) yields

$$0 < \nu((p_v \times p_u) \times p_u)q_w = ((p_v \times p_u) \times p_u)(\lambda p_u - \mu p_v) = \mu \|p_v \times p_u\|^2.$$

Hence also $\mu > 0$.

(iii)$\Longrightarrow$(i): Taking the cross product of (2.E) with p_u, we find that $(p_v \times p_u)$ and $(p_u \times q_w)$ are parallel. Their orientation agrees by (2.I).

♣

Remarks

1 Equation (2.e) can be found e.g. in [Farin '82, p.272], equation (2.E) in [Liu '86 p.437], [Becker '86 p. 225], [Piper p.227, (4.1) '87], [Peters '88], [Degen '89 p.10], and [Liu, Hoschek '89 Thm.1].

2 As Corollary 3.14 and Algorithm 5.I will show, it can be useful to relax the regularity assumption at patch corners. Regularity of the parametrization ensures that the smoothness of the parametrization implies smoothness of the surface (e.g. [do Carmo '76 Prop. 3, p. 63]). Checking smoothness at singular points is more complicated as the examples $t \mapsto (t^2, t^3)$ and $t \mapsto (0, t^3)$ show: both parametrizations are smooth and singular at 0; however, the first curve is not first-order continuous at 0, while the second is. Hence differential geometry restricts its attention to C^k manifolds based on regular parametrizations ([do Carmo '76 p.52],[Klingenberg '83, 3.1.1]).

3 Requiring that the patches are polynomial when parametrized by the parameter of a common boundary curve is convenient, since it keeps the number of constraints on the polynomial coefficients and hence the degree low. In general, one can always reparametrize the patches to match the lowest regular polynomial parametrization of the boundaries.

4 For use in CAGD it is important that the orientation of the patches is controlled since improper orientation allows two patches to join with a cusp; that is, the patches meet in a sharp edge.

5 If p and q are polynomials, then, up to a common factor, so are λ, μ and ν. (See [Liu '86] for a similar claim.) Here are the details.

(2.5) Corollary. [polynomial weight functions] *If p and q are polynomials, then, up to a common factor, λ, μ and ν in Lemma 2.2 are polynomials in u of degree no larger than $\deg(p_v) + \deg(q_w)$, $\deg(p_w) + \deg(q_u)$, $\deg(p_u) + \deg(q_v)$ respectively.*

Proof. Apply Cramer's rule to

$$[p_u, p_v]\begin{bmatrix}\lambda \\ -\mu\end{bmatrix} = \nu q_w \tag{2.E}$$

along the boundary to obtain the (formal) solution

$$\begin{bmatrix}\lambda \\ -\mu \\ \nu\end{bmatrix} = \frac{\nu}{n_k}\begin{bmatrix}l_k \\ m_k \\ n_k\end{bmatrix} \tag{2.6}$$

for some polynomials l_k, m_k and n_k. Since p_u and p_v map into $\mathbb{R}^3$, $\det[p_u, p_v]$ is not welldefined. However, dropping the k^{th} and retaining the i^{th} and j^{th} coordinate, extracts the polynomial $p^{ij} : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ from p. Hence

$$l_k := \det[q_w^{ij}, -p_v^{ij}] \quad m_k := \det[p_u^{ij}, q_w^{ij}] \quad n_k := \det[p_v^{ij}, p_u^{ij}].$$

are welldefined. Since, by (2.I), $p_u(u)$ and $p_v(u)$ are linearly independent for any u, there exists a k such that n_k is a nontrivial polynomial. If each zero u_0 of n_k is also a zero of l_k and m_k so that the common factors can be canceled, then the first two equations of (2.6) are welldefined and not just a formal solution, By (2.I) there exists a $k' \neq k$ such that $n_{k'}(u_0) \neq 0$. Hence a formula like (2.6) with k replaced by k' holds in some neighborhood of u. This implies that l_k/n_k and m_k/n_k are polynomials and that their degree is bounded by the degree of the numerator. The third equation is trivial.

♣

6 In terms of the BB-coefficients, the C^1 common tangent plane constraints amount to setting the coefficients of the function $(p_v \times p_u)q_w$, resp. $\lambda p_u - \mu p_v - \nu q_w$, to zero.

7 The advantage of formulating (2.E) with three weight functions is symmetry. Often the C^1 conditions for abutting polynomial patches are stated in the form

$$q_w = \alpha p_u + \beta p_v \tag{2.7}$$

where α and β are *polynomials* [Bezier '72 p178], [Bezier '86 p42 (163)], [Becker '86 p226], [DeRose '85 p59], [Farin '82 p277], [Farin '83 p50,57], [Farin '88 p248,250], [Faux, Pratt '79 p216], [Hahn '87 p14], [Höllig '86 p14], [Jones '88 p329], [Lin, Hoschek '89 Table 1], [Sabin '77 p85], [Sarraga '86 p5], [van Wijk '84 p4], [Veron, Ris, Musse '79 p269], [Yamaguchi '88 p224]. However, Lemma 2.2 and Corollary 2.5 make clear that α and β are, in general, rational functions (even after removing a common factor), i.e. a polynomial divided by a polynomial. Stipulating that α and β be polynomials is equivalent to restricting either μ or ν to be constant.

(2.8) Example The boundary and first off-boundary layer of BB coefficients of p and q are given by

	$\begin{bmatrix}1\\1\\0\end{bmatrix}$		$\begin{bmatrix}23/6\\13/12\\0\end{bmatrix}$		$\begin{bmatrix}6\\1\\1\end{bmatrix}$		first layer of patch p
$\begin{bmatrix}0\\0\\0\end{bmatrix}$		$\begin{bmatrix}2\\0\\0\end{bmatrix}$		$\begin{bmatrix}4\\0\\1\end{bmatrix}$		$\begin{bmatrix}6\\0\\1\end{bmatrix}$	common boundary
	$\begin{bmatrix}1\\-1\\0\end{bmatrix}$		$\begin{bmatrix}7/3\\-11/12\\1\end{bmatrix}$		$\begin{bmatrix}5\\-1/2\\1\end{bmatrix}$		first layer of patch q

(2.9) Figure: Data along a cubic boundary.

Computing the derivatives as differences between coefficients (with $u = v = w = 0$ corresponding to the zero coefficient) one finds that

$$p_v \sim [\begin{bmatrix}1\\1\\0\end{bmatrix}, 2\begin{bmatrix}11/6\\13/12\\0\end{bmatrix}, \begin{bmatrix}2\\1\\0\end{bmatrix}], \quad p_u \sim [\begin{bmatrix}2\\0\\0\end{bmatrix}, 2\begin{bmatrix}2\\0\\1\end{bmatrix}, \begin{bmatrix}2\\0\\0\end{bmatrix}],$$

$$q_w \sim [\begin{bmatrix}1\\-1\\0\end{bmatrix}, 2\begin{bmatrix}1/3\\-11/12\\1\end{bmatrix}, \begin{bmatrix}1\\-1/2\\0\end{bmatrix}]$$

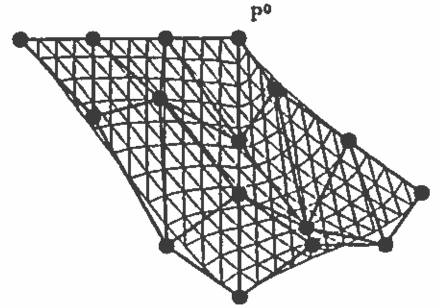
and checks that

$$\lambda := [3/2, 1], \quad \mu := [3/2, 1/2], \quad \nu := [3/2, 1]$$

satisfy $\lambda p_u = \mu p_v + \nu q_w$. None of the weight functions is of the form $[1,1]^k$, i.e. none just raises the degree of p_u, p_v or q_w, and $\alpha = \lambda/\nu = 1$ and $\beta = \mu/\nu$ is not a polynomial proving that a polynomiality assumption for (2.7) is too restrictive. As an aside, we note that $\det[p_u, p_v, q_w] = 0$ but

$$\det[\begin{bmatrix}2\\0\\1\end{bmatrix}, \begin{bmatrix}11/6\\13/12\\0\end{bmatrix}, \begin{bmatrix}1/3\\-11/12\\1\end{bmatrix}] = \frac{1}{8} \neq 0,$$

i.e. the cross boundary quadrilaterals of BB-coefficients are not all coplanar.



(2.10) Figure: The two cubic patches of (2.8) meet in a nonconstant C^1 match along P^0P^1. The 10+10-4 coefficients of p (left) and q (right) are connected to form the joint BB-net. The shading is computed for each of the small triangles.

The polygonal shading in Figure 2.10 reveals a well-behaved surface whose only interesting feature are the curved isoparametric lines. ♣

[Liu '86 p438] proposes constant coefficient matches for the surface construction, but additionally gives an example of a surface continuation based on non-constant weight functions. Non-constant constructions appear in [Chiyokura, Kimura '83 p295], [Piper '86 p227] and [Peters '88b]. Using the notation introduced in [Peters '88b], C^1 matches can be characterized by the triple (degree of λ, degree of μ, degree of ν). Piper's construction is a (2,1,1)-match, and Chiyokura and Kimura achieve a (1,1,1)-match by restricting the transversal derivatives to quadratics.

2.1 Second-order continuity between two patches

Assume that p and q meet C^1. Then p and q form a C^2 surface if and only if, at each point of the boundary, the derivative of the normal map of p is welldefined and agrees with the derivative of the normal map of q [cf. do Carmo '76 p136]. Since

$$0 = D_i(N D_j p) = (D_i N)(D_j p) + N(D_i D_j p) \quad \text{for } i,j \in \{1,2\},$$

and since D_i is a linear map with $D_i N \perp N$, this is equivalent to the existence of a symmetric bilinear map II from the tangent plane into itself such that, at each point of the boundary,

$$-N(D_i D_j p) = (D_i p) II (D_j p) \quad \text{and} \quad -N(D_i D_j q) = (D_i q) II (D_j q)$$

for $i,j \in \{1,2\}$.

(2.11) Lemma. *Two patches p and q that share a common boundary curve with parameter u and meet C^1 with weight functions λ, μ and ν form a C^2 surface if and only if, at each point of the boundary,*

$$N(\nu^2 q_{ww} - \lambda\nu q_{wu}) = N(\mu^2 p_{vv} - \lambda\mu p_{uv}). \tag{2.12}$$

Proof. Assume II is defined by the patch q, i.e. by

$$-Nq_{uu} = q_u II q_u, \quad -Nq_{uw} = q_u II q_w, \quad -Nq_{ww} = q_w II q_w.$$

Since the match is C^0, $p(u,0) = q(u,0)$, and hence

$$-Np_{uu} = p_u II p_u.$$

Since $Np_v = Nq_w = Np_u = 0$ and since the match is C^1

$$0 = N(\mu p_v + \nu q_w - \lambda q_u)_u$$
$$= N\mu p_{uv} - \nu q_w II q_u + \lambda q_u II q_u$$
$$= N\mu p_{uv} + \mu p_v II p_u,$$

i.e. $-Np_{uv} = p_v II p_u$. Thus only $-Np_{vv} = p_v II p_v$ has to be enforced. This constraint can be rewritten with the help of the C^0 and the C^1 constraints:

$$-\mu^2 Np_{vv} = (\mu p_v) II (\mu p_v)$$
$$= (-\nu q_w + \lambda q_u) II (-\nu q_w + \lambda q_u)$$
$$= (\nu q_w) II (\nu q_w) - (\nu q_w) II (\lambda q_u) + (\lambda q_u) II (\mu p_v)$$
$$= -\nu^2 N q_{ww} + \lambda\nu N q_{uw} - \lambda\mu N p_{uv}.$$

♣

(2.13) Corollary. *Two patches p and q that share a common boundary curve with parameter u and meet C^1 with weight functions λ, μ and ν form a C^2 surface if and only if, there exist scalar valued weight functions α, β, γ and $\delta \neq 0$ (of u) such that at each point of the boundary,*

$$\delta((\nu^2 q_{ww} - \lambda\nu q_{wu}) - (\mu^2 p_{vv} - \lambda\mu p_{vu})) = \alpha p_u + \beta p_v - \gamma q_w \qquad (2.14)$$

(2.15) Example Consider a polynomial surface p subdivided along $v = 0$ into two patches $p_{|\square}$ and $p_{|\Delta}$, where $\square$ is the unit square with parameters u and v and Δ is an abutting unit triangle with parameters u and w. Then the C^1 constraints, $\lambda p_u = \mu p_v + \nu p_w$, imply that $\lambda u = \mu v + \nu w$ and hence

$$\nu^2 q_{ww} - \lambda\nu q_{wu} - \mu^2 p_{vv} + \lambda\mu p_{uv}$$
$$= \nu^2 p_{ww} - \nu p_{w,\mu v+\nu w} - \mu^2 p_{vv} + \mu p_{\mu v+\nu w,v}$$
$$= -\nu\mu p_{wv} + \mu\nu p_{wv} = 0.$$

$p_{vv} = \nu^2 q_{ww} - \lambda\nu q_{wu} + \lambda p_{vu} + \ldots$

Chapter Three

The vertex enclosure problem

Fitting polynomial patches smoothly to a mesh of curves is not in general a well-posed problem regardless of the degree (and shape) of the patches used: not every mesh with a welldefined tangent plane at the mesh points has a *smooth regularly parametrized interpolant with one patch per mesh facet*. This chapter derives the necessary and sufficient *vertex enclosure constraint* on a mesh of polynomial curves that guarantees the existence of a regular smooth interpolant and offers a sufficient constraint that can easily be checked: if all mesh curves match the same second-order data at the mesh points, then a smooth interpolant exists (see Theorem 3.6).

For blended interpolants, it is well known that the existence of a welldefined tangent plane at the data points does not guarantee the existence of a C^1 mesh interpolant (see e.g. [Gregory '74], [Nielson '79]). In particular, since the blending approach requires knowledge of normal derivatives along the boundaries, p_{uv} and p_{vu} are given independently at any point P. Thus, if the patches are polynomials and P has n neighbors, the data must satisfy n additional vector constraints ($= 3n$ constraints) of the type $p_{uv} = p_{vu}$ at P. However, as [Sarraga '86] and [Watkins '88] point out, curve meshes by themselves need only satisfy one vector constraint per even-point and none at odd-points to permit solving for the mixed derivatives. Due to its circulant structure, the corresponding matrix is always invertible at odd-points, but rank-deficient at even-points. (A similar parity phenomenon has also troubled the schemes of [van Wijk '83] and [Peters '88a].) For 4-points, Sarraga exhibits a sufficient constraint, Constraint 3.1(a) below, (see also [Bezier '86 (187)]) that forces the right hand side into the range of the rank-deficient matrix. Unfortunately, the analysis and construction apply only to 4-points.

This is where Lemma 3.4 comes into play. It applies the analysis of Sarraga and Watkins to arbitrary combinations of total degree and tensor product patches of possibly differing degree. The central Theorem 3.6 then gives the precise necessary and sufficient constraint that guarantees that a smooth regularly parametrized surface with one patch per mesh facet can be fit to a mesh of curves.

(3.1) Constraint. [sufficient vertex enclosure constraints] *At every even-point P either of the following holds.*

(a) [collinearity constraint] *All odd-numbered and all even-numbered curves emanating from P have collinear tangent vectors; that is, the tangent vectors form an 'X'.*

(b) [curvature constraint] *The mesh curves emanating from P are compatible with some second fundamental form at P.*

Constraint 3.1(a) applies only to 4-points and is due to [Bezier '86 p.47] and [Sarraga '87]. Constraint 3.1(b) is valid more generally. It requires that the maximal and the minimal curvature and one principal direction are welldefined at P. Note that both constraints are satisfied by a regular (tensor product) mesh. Example 3.10 shows that Constraint 3.1 is sufficient, but not necessary. Even though the analysis is carried out in the BB-representation, it applies to any patches that are twice continuous differentiable at the vertex.

The chapter also offers three techniques to enclose a vertex if the data do not satisfy the vertex enclosure constraint.

1 Use of rational patches whose denominator vanishes at all data points.

2 Use of the splitting-and-averaging construction.

3 Singular parametrization of the boundary curves at the vertex.

3.1 Derivation of the mixed-derivative constraints

By Lemma 2.2 and Corollary 2.5, the C^1 constraints can be expressed as

$$f(u) := \lambda p_u - \mu p_v - \nu q_w \stackrel{!}{=} 0$$

where λ, μ and ν are, without loss of generality, polynomials. The gist of the vertex enclosure problem can be captured by looking at $f'(0) = 0$, i.e.

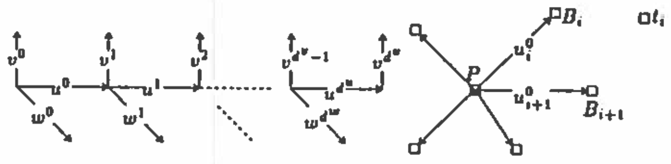
$$\mu p_{uv} + \nu q_{uw} = \lambda p_{uu} + (\lambda_u p_u - \mu_u p_v - \nu_u q_w) \qquad (2.E_u)$$

at a data point P. The data are ***admissible***, i.e. the vertex enclosure constraint is met, if the system of constraints that arises from collecting $(2.E_u)$ for each mesh curve emanating from P is solvable.

We look at $(2.E_u)$ in more detail. For mnemonic efficiency, the BB-coefficients of p_u, p_v and q_w are denoted by u^i, v^i and w^i. That is, u^i, v^i and w^i are difference vectors of Bézier control points as sketched in Figure 3.2.

Thus the relevant terms of (2.E) for the analysis of $(2.E_u)$ are .

$$\underbrace{[\lambda^0, d^\lambda \lambda^1, \ldots]}_{\lambda}\underbrace{[u^0, d^u u^1, \ldots]}_{p_u} =$$
$$\underbrace{[\mu^0, d^\mu \mu^1, \ldots]}_{\mu}\underbrace{[v^0, d^v v^1, \ldots]}_{p_v} + \underbrace{[\nu^0, d^\nu \nu^1, \ldots]}_{\nu}\underbrace{[w^0, d^w w^1, \ldots]}_{q_w}, \qquad (2.E')$$



(3.2) Figure: Notation for the enclosure problem: the difference vectors of a tensor product and of a total degree patch along the shared edge (left) and the BB-coefficients labeled around the data point P (right).

where $d^\lambda + d^u = d^\mu + d^v = d^\nu + d^w$. We need not worry about the meaning of the λ^1, u^1, μ^1, etc. for degree zero polynomials, since then $d^\lambda = d^u = \ldots = 0$. The C^1-constraints are obtained by setting each coefficient (with respect to the BB-basis) to zero. This yields the *tangent constraints*

$$\lambda^0 u^0 = \mu^0 v^0 + \nu^0 w^0 \qquad (2.E_0)$$

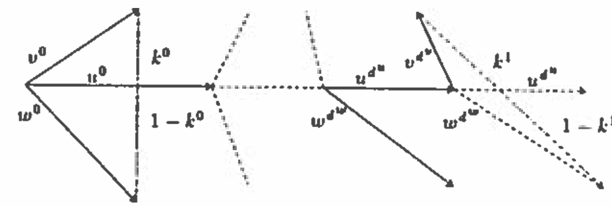
and the *twist constraints*

$$d^u \lambda^0 u^1 + d^\lambda \lambda^1 u^0 = d^v \mu^0 v^1 + d^\mu \mu^1 v^0 + d^w \nu^0 w^1 + d^\nu \nu^1 w^0. \qquad (2.E_1)$$

Now consider the n mesh curves emanating from P with patches and curves labeled clockwise and cyclically. If we set $\lambda_i^0 := \alpha_i \eta_i$, $\mu_i^0 := \alpha_i k_i$ and $\nu_i^0 := \alpha_i(1-k_i)$ with $\alpha > 0$ (to comply with (2.1)), then η_i and k_i are uniquely determined by

$$\eta_i u_i^0 = k_i v_i^0 + (1-k_i) w_i^0 \quad i \in \{1, \ldots, n\}. \qquad (2.E_0')$$

Since the boundary curves are given, u_i^0, v_i^0 and w_i^0 are fixed (and coplanar), and the u_i^1 are fixed (but need not lie in the same plane). Only the vectors v_i^1 and w_i^1 and the scalars α_i, λ_i^1, μ_i^1 and ν_i^1 are off-hand still free.



(3.3) Figure: Geometric meaning of k^0 and k^1.

Remarks

1 We say 'off-hand' since the degree of the weight functions may be so low that λ_i^1, μ_i^1 and ν_i^1 are already fixed by the tangent constraints at the neighbor point P_i. Similarly, if the degree of p_v or q_w is low, then v^1 and w^1 can be pinned down or shared by overlapping twist constraints as in the case of total degree cubics. Hence, we assume in the following that the patches are at least bicubic or quartic, so that the twist constraints are separated.

2 We avoid notational problems that arise, e.g. in the analysis of cubic-by-sextic patches, by raising the degree of any two adjacent boundary curves, p_i and p_{i+1} to the same value $d_i := \max\{\deg(p_i), \deg(p_{i+1})\}$. Cubic-by-sextic patches thus become bisextic patches, and the interior Bézier coefficient of the i^{th} patch and closest to P is now welldefined. This coefficient is called the *twist coefficient* (for historical reasons) and is denoted by t_i (see Figure 3.2).

The comparison of the degrees of freedom with the number of equations then suggests the following approach: fix the scalar variables, λ_i^1, μ_i^1, and ν_i^1, by some rule and satisfy the n constraints with the n twist coefficients. That is, solve

$$k_i d_{i-1} t_{i-1} + (1-k_i) d_i t_i = r_i, \qquad (2.E_1')$$

where

$$\alpha_i r_i := (d_i^u \lambda_i^0 u_i^1 + d_i^\lambda \lambda_i^1 u_i^0 +$$
$$\mu_i^0(d_{i-1}B_i + (d_{i-1} - d_i^v)v_i^0) - d_i^\mu \mu_i^1 v_i^0 +$$
$$\nu_i^0(d_i B_i + (d_i - d_i^w)w_i^0) - d_i^\nu \nu_i^1 w_i^0)$$

and $B_i := u_i^0 + P$ as shown in Figure 3.2. (Note that r_i simplifies considerably if all patches have the same degree at the outset.) The constraints at P form one n by n system for each coordinate, namely

$$KT = R, \tag{3.T}$$

where

$$K := \begin{bmatrix} d_1k_1 & d_2(1-k_1) & \dots & 0 & 0 \\ 0 & d_2k_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \dots & d_{n-1}k_{n-1} & d_n(1-k_{n-1}) \\ d_1(1-k_n) & 0 & \dots & 0 & d_nk_n \end{bmatrix},$$

$$T := \begin{bmatrix} t_1 \\ \vdots \\ t_n \end{bmatrix} \text{ and } R := \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}.$$

Unfortunately, as Lemma 3.4 below shows, (3.T) is not always solvable since the contribution of the u^1 terms in R is arbitrary and $\lambda_i^0 \neq 0$ in general (cf. also Claim 6.7).

(3.4) Lemma. [the parity phenomenon] *The matrix in (3.T) is of full rank if and only if P is an odd-point. Otherwise its rank is deficient by one.*

Proof. The first $n-1$ rows and columns of K form an upper triangular matrix with non-zero diagonal. Hence the rank is at least $n-1$. For any n-vector $(t_i)_{i=1,\dots,n} \in \ker K$

$$k_i d_i t_i + (1-k_i)d_{i+1}t_{i+1} = 0,$$

and thus

$$\left(\prod_{i=1}^{n} \frac{k_i d_i}{(k_i - 1)d_{i+1}}\right) t_1 = t_1. \tag{3.5}$$

Looking at the u_i^0 as 2-vectors in the tangent plane, (2.E$_0'$) implies for the i^{th} edge (with cyclic count)

$$k_i = \frac{\det[u_i^0, u_{i+1}^0]}{\det[u_i^0, u_{i+1}^0 - u_{i-1}^0]}$$

and hence

$$\prod_{i=1}^{n} \frac{k_i d_i}{(k_i - 1)d_{i+1}} = (-)^n.$$

That is, $(-1)^n t = t$ for every element t of the kernel of K. Hence the kernel is nontrivial if and only if n is even. ♣

3.2 The vertex enclosure constraint

Lemma 3.4 makes clear that the only hope for solving (2.E$_u$) at even-points is to adjust the parametrization, i.e. to adjust the scalar weight coefficients so that the right hand side lies in the range of K. The net effect of treating the scalar coefficients as additional variables (and insisting on regularity) is a reduction from one vector constraint (Lemma 3.4) to one scalar constraint. While the precise constraint is not very appealing, Constraint 3.1 is sufficient and can easily be checked. Note that even though Theorem 3.6 is only concerned with the *existence* of a regularly parametrized twice continuously differentiable interpolant, the proof is constructive provided the degree of the weight functions and patches is sufficiently high.

(3.6) Theorem. [sufficiency of C^2 data] *At any even-point P, the normal curvature of the mesh curves emanating from P must be constrained to allow for an interpolating C^1 surface with one regularly parametrized C^2 patch per mesh facet. Either condition of Constraint 3.1 guarantees solvability.*

Proof. We first restrict attention to the λ_i^1 terms, i.e. the derivative of the versal weight function, and show later that the analysis remains unchanged if all weight functions are allowed to vary. Let u_i^x be the first component of u_i^0, $R'^x(t)$ the first component of $(r_i - \frac{1}{\alpha_i} d_i^\lambda \lambda_i^1 u_i^0)$, L the n-vector corresponding to the λ_i^1, $d_i^\lambda > 0$, and

$$U^x := \begin{bmatrix} -\frac{d_1^\lambda}{\alpha_1}u_1^x & 0 & \dots & 0 \\ 0 & -\frac{d_2^\lambda}{\alpha_2}u_2^x & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -\frac{d_n^\lambda}{\alpha_n}u_n^x \end{bmatrix}. \tag{3.7}$$

Then (2.E$_u$) becomes

$$\begin{bmatrix} K & 0 & 0 & U^x \\ 0 & K & 0 & U^y \\ 0 & 0 & K & U^z \end{bmatrix} \begin{bmatrix} T^x \\ T^y \\ T^z \\ L \end{bmatrix} = \begin{bmatrix} R'^x \\ R'^y \\ R'^z \end{bmatrix}. \tag{3.T'}$$

We show that (3.T') is not always solvable. To simplify the analysis, the coordinate system is transformed rigidly and orientation preserving so that P is mapped to the origin and the normal N at P to $(0, 0, 1)$. This leaves the basic block structure of (3.T') unchanged:

$$\begin{bmatrix} K_1 & 0 & 0 & U_1 \\ 0 & K_2 & 0 & U_2 \\ 0 & 0 & K_3 & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ L \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}. \tag{3.T''}$$

Since all the u_i^0 lie in the same tangent plane, the transformation creates a 0-matrix in the lower right hand corner. On the other hand, the partial system corresponding to the first $2n$ equations is of full rank since there are exactly two linearly independent vectors among the u_i^0 and we may choose the degree of λ large enough so that λ^1 can be varied freely. Hence, we need only consider $K_3T_3 = R_3$. The only contributions to R_3 come from the nontangential component, $\lambda^0 u_{i3}^1$ (cf. (2.E$_1'$)). All other vectors lie the tangent plane and hence appear only in R_1 and R_2. Since K_3 is of rank $n-1$, (3.T'') is solvable exactly when R_3 is in the span of the first $n-1$ columns of K_3; that is, if and only if, the following **vertex enclosure constraint** holds:

$$\det \begin{bmatrix} d_1k_1 & d_2(1-k_1) & \dots & 0 & \lambda_1^0 u_{13}^1 \\ 0 & d_2k_2 & \dots & 0 & \lambda_2^0 u_{23}^1 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \dots & d_{n-1}k_{n-1} & \lambda_{n-1}^0 u_{n-1,3}^1 \\ d_1(1-k_n) & 0 & \dots & 0 & \lambda_n^0 u_{n3}^1 \end{bmatrix} = 0. \tag{3.8}$$

Constraint 3.1(a) implies (3.8) by stipulating that all λ_i^0 be zero. To see the sufficiency of Constraint 3.1(b), we look at the nontangential component of (2.E$_u$). Since the mesh curves match second-order data at P, there exists a symmetric matrix II of rank two such that $Np_{uu} = -p_u II p_u$, namely the second fundamental form (see e.g. [Faux,Pratt '79]). Hence, the nontangential component of (2.E$_u$) at $u = 0$ is

$$-\lambda p_u II p_u = \mu N p_{uu} + \nu N q_{uw}. \qquad (2.E_u^N)$$

By choosing the normal components of the t_i so that all p_{uv} satisfy $Np_{uv} = -p_u II p_v$, and hence $Nq_{uw} = -q_w II q_u = -q_w II p_u$, at the data point, equation (2.E$_u^N$) reduces to

$$(\lambda p_u - \mu p_v - \nu q_w) II p_u = 0 \qquad \text{at} \qquad u = 0.$$

But this is already implied by the tangent constraints (2.E$_0$).

Including all scalar variables of (2.E$_1$) into the analysis will not change the result, since the corresponding vectors u_i^0, v_i^0 and w_i^0 all lie in the same (tangent) plane. ♣

To obtain meshes that satisfy the vertex enclosure constraint, consider the standard case $d_1 = \ldots = d_n$ and rewrite (3.8) as

$$\lambda_n^0 u_{n3}^1 = \sum_{j=1}^{n-1} (-)^{j+1} \left(\prod_{l=1}^{j} \frac{1 - k_{l-1}}{k_l} \right) \lambda_j^0 u_{j3}^1. \qquad (3.8')$$

Since α is restricted in sign, any change in the λ_i^0 or k_i has to come from a perturbation of the tangent vectors, and hence will affect many other terms. However, there are a number of strategies for perturbing a given mesh of curves to satisfy (3.8).

1 Adjust one of the mesh curve, e.g. choose u_{n3}^1 to satisfy (3.8′). Besides lack of symmetry, any change in u_{n3}^1 has also an impact on the equations at P_n if the degree of the boundary parametrization is no larger than three.

2 Enforce Constraint 3.1(b) locally and symmetrically by locally interpolating positional, normal and curvature data. The references [Sabin '68], [de Boor, Höllig, Sabin '87] and [Peters 88c] show under what conditions local Hermite interpolation by cubic space curves is possible.

3 If the mesh of data is regular, use global cubic spline interpolation. This enforces both Constraint 3.1(a) and 3.1(b).

4 Use mesh curves of higher degree.

The second issue of interest is whether Constraint 3.1 is, at least for 4-points, necessary. For this, recall that the second fundamental form II at a data point P consists of three pieces of information, namely the two extremal curvatures and a principal direction. If, at a 4-point, at least one of the two pairs of opposing tangents is not collinear, then three of the mesh curves can be used to define II, while the fourth curve has to satisfy a single scalar constraint. Surprisingly, even though this constraint implies the vertex enclosure constraint (3.8) and acts on the same variables as (3.8), it is not equivalent to (3.8).

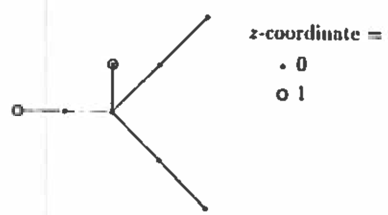
(3.9) Claim. *The existence of a solution to the compatibility problem at a 4-point does not imply Constraint 3.1.*

We prove the claim by exhibiting data that can be extended to a set of admissible data, but satisfy neither the collinearity nor the curvature condition of Constraint 3.1.

(3.10) Example Consider the four quadratic curve segments emanating from 0 with first differences

$$[u_1^0, u_1^1] := \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad [u_2^0, u_2^1] := \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix},$$

$$[u_3^0, u_3^1] := \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 0 & 0 \end{bmatrix}, \quad [u_4^0, u_4^1] := \begin{bmatrix} -1 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$



(3.11) Figure: Four curve segments.

The normal at P is $N = (0,0,1)$. Constraint 3.1(a) does not hold. For Constraint 3.1(b) to hold there must exist

$$H := \begin{bmatrix} a & b & 0 \\ b & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

such that $Nu_i^1 = -u_i^0 H u_i^0$ for $i \in \{1,2,3,4\}$. However,

$$Nu_1^1 = Nu_1^4 = 1, \quad Nu_1^2 = Nu_1^3 = 0,$$

and this implies that $a = c = -1$ and $b = 2$ and $b = -2$. Hence Constraint 3.1(b) does not hold. The vertex enclosure constraint (3.8), however, is

satisfied since

$$\det \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = 0.$$

♣

3.3 Implications of the vertex enclosure constraint

The proofs of Lemma 3.4 and Theorem 3.6 do not depend on the patch type.

(3.12) Corollary. [independence of patch type] *The vertex enclosure problem exists for any combination of 3- or 4-sided or even n-sided patches.*

Rational patches of the form R/r, where R is a vector-valued and r a scalar-valued polynomial, help (only) if they destroy second-order smoothness of the patches at the data points.

(3.13) Corollary. [rational patches] *The vertex enclosure problem exists for rational polynomial patches unless the polynomial in the denominator vanishes at all data points.*

Proof. Gregory's patches (cf. [Gregory '74], [Chiyokura '83]) are rational with vanishing denominator and match arbitrary cross derivatives at the data points. Conversely, rational patches are only useful if r must vanishes at the data points. For, if $r(0) \neq 0$, one may assume, after scaling and shifting, that $r(0) = 1$ and $R(0) = 0$, after which the common-tangent-plane constraint (2.E) reads

$$\lambda\frac{R_v r + r_v R}{r^2} - \mu\frac{R_v r + r_v R}{r^2} - \nu\frac{Q_w q + q_w Q}{q^2} \stackrel{!}{=} 0.$$

Differentiation with respect to u yields

$$\lambda_u\frac{R_v r + r_v R}{r^2} + \lambda\left(\frac{R_{uv} r + 2r_u R_v + r_{uv} R}{r^2} - 2r_u\frac{R_v r + r_v R}{r^3}\right)$$

$$= \lambda_u R_v + \lambda R_{uv} =$$

$$\mu_u R_v + \mu(R_{vu} + r_v R_u - r_u R_v) + \nu_u Q_w + \nu(Q_{wu} + q_w R_u - r_u Q_w),$$

i.e.

$$\mu R_{vu} + \nu Q_{wu} = \lambda R_{uu} + (\text{tangent vectors})$$

This is, however, the same setup discussed in Lemma 2.2 and Theorem 3.6. ♣

A second look at (2.E) for rational patches shows that the high degree of the polynomials in the numerator and denominator is inherent in the construction. (Gregory's 4-sided patches are bisceptic over biquartic and his 3-sided patches are septic over cubic.) If, for example, the weight functions μ and ν are constant, then the number of *vector* constraints increases to $d^R + d^r$ as opposed to d^R in the polynomial case since the degree of the numerator is $d^R + d^r - 1$. Yet there are only d^r additional *scalar* degrees of freedom.

So far, we assumed that the boundary curves are regularly parametrized. However, as Algorithm 5.II demonstrates, singularly parametrized surfaces can be can smooth and well-behaved and the following Corollary is worth noting.

(3.14) Corollary. [singular parametrizations] *The vertex enclosure problem can be avoided by parametrizing the boundary curves singularly at the vertex.*

Proof. Denote the k^{th} derivative of p along the edge corresponding to the i^{th} boundary curve by $D_i^k p$. Reparametrize the boundary by $p \circ \phi$ so that

$$D_i(p \circ \phi)|_0 = 0 \quad \text{and} \quad D_i^2(p \circ \phi)|_0 = D_i p|_0 \quad .$$

Then the new tangent coefficients have the same coordinates as the vertex and hence the 'tangent constraints' (2.E$_0$) are satisfied for any choice of λ_i^0. In particular, $\lambda_i^0 = 0$ and $t_i = P$ solve (2.E$_1$). The tangent plane is now defined by the second-difference vectors and these are in agreement with the first-difference vectors of the original regular parametrization. ♣

Corollary 3.14 was motivated by the following example. **(3.15) Example** Consider the data displayed in Figure 3.16. The parabola $c_1(t) := (0, t, t^2)$ with normal $(0, -2t, 1)$ and the straight line $c_2(s) := (s, s, 0)$ with normal $(0, 0, 1)$ are to be interpolated by a smooth surface. Since

$$\frac{\partial}{\partial t}N(0,0) = (0, -2, 0) = -2(Dc_1)(0,0),$$
$$\frac{\partial}{\partial s}N(0,0) = (0,0,0) = 0(Dc_2)(0,0),$$

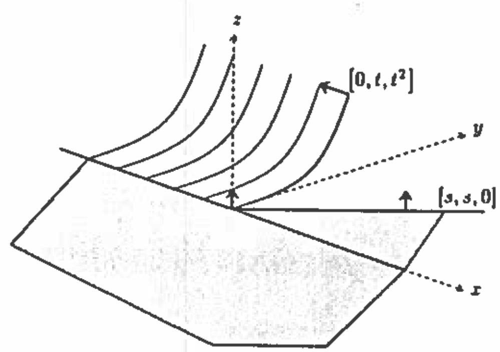
the principal directions of any interpolating patch coincide at the origin with the tangents of the boundary curves. Since the principal directions, $(0,1,0)$ and $(1,1,0)$, are not perpendicular, while the principal curvatures, -2 and 0, are not equal, no interpolating patch can be a part of a C^2 manifold [Klingenberg '83, 3.5.3]. Looking at the data as a variation of the curve data of Example 3.10 with $u_4^1 := (-1,0,0)$, the vertex enclosure constraint reads

$$\det \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = -2 \neq 0.$$

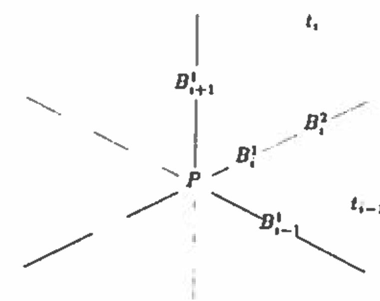
However, the approach of [Bajaj,Ihmsung '89] yields the implicitly defined family of singular surfaces $F(x,y,z) = 0$, where

$$F(x,y,z) = -c_3 z^2 + c_3 y^2 z - c_2 yz + c_4 x^2 z + (3c_2 + c_1)xz/2$$
$$+c_2 y^3 - (3c_2 + c_1)xy^2/2 + c_1 x^2 y - (c_1 - c_2)x^3/2.$$

The choice $c_1 = -c_2 = c_3 = c_4 = 1$ shows that a smooth interpolant exists, even though it cannot be regularly parametrized in the parameters of the boundary curve. ♣



(3.17) Figure: Data that cannot not be C^1 interpolated by a C^2 patch.



(3.18) Figure: Notation for splitting-and-averaging.

Covering an n-sided domain by n non-overlapping patches, i.e. splitting the domain into n triangles, and computing the additional boundary curves by averaging, is another technique that resolves the compatibility problem - even though it creates even-points. A partial explanation is provided by observing that the *splitting-and-averaging* technique generates the coefficients of the new interior boundaries based on the twist coefficients rather than *vice versa* (cf. [Farin '83], [Piper '87], [Shirman, Séquin '87], [Peters '88b]). The other ingredient is the symmetry of the averaging construction; that is, of the rules

$$B_i^1 \leftarrow \mathrm{avg}(P, B_{i-1}^1, B_{i+1}^1, \mathrm{sds}(f_i))$$
$$B_i^2 \leftarrow \mathrm{avg}(B_i^1, t_{i-1}, t_i, \mathrm{sds}(f_i)),$$

where $\mathrm{avg}(p1,p2,p3,i) := p1 + \frac{1}{\lambda}(p2 + p3 - 2p1)$, $\lambda := 2(1 - \cos(\frac{2\pi}{i}))$ and $\mathrm{sds}(f_i)$ is the number of edges of the i^{th} original facet (cf. Section 5). Note that if $\mathrm{sds}(f_i) = 3$, as in the Clough-Tocher split, then $\lambda = 3$ and hence

$$B_i^1 = (P + B_{i-1}^1 + B_{i+1}^1)/3.$$

(3.19) Corollary. [splitting-and-averaging] *Splitting-and-averaging leads to admissible data.*

Proof. Assume, without loss of generality, that the degree of all patches is the same, say d. Since the twist coefficients, t_i, can be chosen to enforce (2.E$_1$) at each original (second) edge, it is sufficient to check that (2.E$_1$),

$$dk_i t_{i-1} + d(1-k_i)t_i = r_i \qquad (2.\text{E}_1)$$

holds for fixed t_i and t_{i-1} at each new (splitting) boundary. By the averaging construction, $k_i = 1 - k_i = 1/2$, $\alpha = 1$, $\mu = \nu = 1/2$, hence $d^\lambda = d^\mu = d^\nu = 0$ and

$$2r_i/d = \lambda_i^0 u_i^1 + 2B_i^1$$

for any new boundary. On substituting this into (2.E$_1$) and dividing by $d/2$, the constraint becomes

$$t_{i-1} + t_i = \lambda_i^0 u_i^1 + 2B_i^1. \qquad (2.\text{E}_1')$$

But this is implied by the construction $B_i^2 \leftarrow \text{avg}(B_i^1, t_{i-1}, t_i, \text{sds}(f_i))$. ♣

We note that rational patches and splitting-and-averaging work because they generically destroy second-order smoothness at the data points.

Chapter Four

Local Smooth Surface Interpolation: a Classification

The table below classifies some algorithms in the literature that construct smooth, interpolating surfaces with piecewise polynomials, i.e. algorithms that convert a 'mesh of 3D data' into the coefficients of a piecewise polynomial representation. The 'data' column describes the interpolation conditions in more detail: upper case letters indicate the highest-order data matched by the interpolant; lower case letters signal important restrictions on the data. The 'edges' column specifies how many edges a facet of the mesh may have; if the entry is k, an arbitrary number of edges can be handled by the algorithm. Tensor-product patches are prefixed 'bi' in the 'degree' column. The degree of a denominator polynomial (if any) is preceded by a '/'. The 'weights' column, finally, details how the algorithms sew the polynomial patches into a smooth quilt. All entries are explained in more detail below. The algorithms are grouped into blocks according to the three major approaches to the interpolation problem: the single patch approach, the blending approach, and the splitting approach.

A regular parametric piecewise polynomial surface is *smooth* or C^1 ($= G^1 = GC^1 = V^1 = VC^1$) if its pieces (patches) join with **oriented tangent plane continuity**; that is, if the surface normals of abutting patches are uniquely defined and agree at every point of the boundary. C^1-smoothness for (bivariate) functions differs from C^1-smoothness for surfaces, since the former is an attribute of one map from the plane to 1-space while the latter is a property of the image of several maps from the unit square or triangle to 3-space.

data	edges	degree	weights	reference
Pr	4	bi4	1,0,0	Bee '86
M	4	bi6	4,3,0	Béz '79
				Sar '86
M	3,4	4,bi4	3,1,1or2,1,0	Pet '89a
Hr	4	bi3	$p_u(N_0 \times N_1)$	Sab '68
Nr	3,4	3,bi3	$(1-t)N_0 + t\omega N_1$	Pet '88a
Dfc	4	bi3		Coo '67
Df	3	2(3+3)	(1/1)	Bar '73
Df	4	bi4/1		Gre '74
M	4	bi4/1	1,1,1	Chi '83
M	3	1+2	[4/2]1,1,1	Her '85
Dc	5	3+2	[6/6](1/1)	Cha '84
Kc	k	3+4/8	$[3(k-2)/3(k-2)](1/1)$	Gre '89
D	3	3+3	[2/2](1/1)	Nie '86
K	3	7+5	[4/4](1/1)	Hag '89
Tf	3	3	0,0,0	Clo '65
Tf	3_6	2	0,0,0	Pow '77
N	3	4	1,0,0	Far '83
T	3	4	2,1,1	Pip '87
M	3,4	4	1,1,1	Shi '87
M	3,4	3	2,1,1	Pet '88b
			1,1,1	Pet '88b
M	$k > 4$	4	3,2,2	Pet '90a
			1,1,1	Pet '90a
N	k	5	1,0,0	Jon '88
N	$3_{1.5}$	bi3	1,0,0	Pet '89b

(4.1) Table: Local Smooth Surface Interpolation with Piecewise Polynomials

(4.1) Table ctd.: P = data point, N = normal, T = tangent, H = curvature, M = curve mesh, D = transversal derivative on mesh, K = transversal curvature on mesh (surface is C^2), (K$\Longrightarrow$ D$\Longrightarrow$ M$\Longrightarrow$ T$\Longrightarrow$ N $\Longrightarrow$ P); f ~ cannot model closed surfaces, c ~ compatibility assumed, r ~ N or H is restricted, $_{1.5}$ ~ 1.5 patches per facet, $_6$~6 or 12 patches per facet.

To appreciate the twofold role of interpolation conditions and the meaning of the 'c' entries in the first column, assume for the moment that no data other than the location of the mesh nodes (data points) are prescribed. Then (2.E) and (2.1), $\lambda p_u = \mu p_v + \nu q_w$, $p_v \times p_u \neq 0$ and $\mu\nu > 0$, form a *nonlinear system of equality and inequality constraints* in the coefficients of p, q, λ, μ and ν. E.g. if the patches are represented in BB-form, the first layers of the BB-net along the patch boundaries and around each data point are connected via a *global* system. Hence, an important feature of any surface construction is the selection of (geometrically meaningful) coefficients to be fixed *a priori* (input or derived from the data), so as to arrive at a sufficient and consistent sequence of local and linear constraints. Such input can range from prescribing a patch complex around the p to be constructed ([Gregory, Hahn 89]), i.e. prescribing q in (2.E), to just fixing the tangent plane at the data points. While additional interpolation requirements can reduce the algebraic complexity of the task, such data must be both available (or easy to generate) and consistent with the polynomial representation. Consistency is a major problem, since each mesh curve, whether input or constructed as a part of the algorithm, carries second-order data in the nontangential component of its second derivative. At the data (mesh) points such data from different curves come together and thus give rise to the **vertex enclosure problem** (cf. Chapter 3). If the data point has an odd number of neighbors, then the data can be dealt with (a certain circulant matrix has full rank);

but if the number of neighbors is even and the impinging patches are at least second-order smooth, then the data have to be constrained (now the aforementioned matrix is rank-deficient). This amounts to one scalar constraint per data point and can be enforced by making the mesh curves match second-order data at the point. If transversal derivatives are prescribed in addition to the mesh curves as in the blending approach, then each patch boundary prescribes the mixed derivative at the data point independently, thus leading to $3k$ constraints per point (one vector constraint for each patch corner). Each of the three major approaches below offers a different solution to the vertex enclosure problem.

An algorithm that constructs just one polynomial patch per mesh facet follows the **single patch approach**. This amounts to interpolation at the mesh nodes, followed by a consistent construction of the mesh curves. For [Sabin '68] and [Peters '88a] the characterization of the weight function is replaced by the definition of the normal direction n along the mesh curves, since both algorithms are based on the following, slightly different, but equivalent scalar reformulation of (2.E): $np_u = np_v = nq_w = 0$. Once n is prescribed, the corresponding constraint system becomes linear and local.

The **blending approach** constructs k polynomial pieces for a k-sided mesh facet. Each piece matches a part of the data, so that a convex combination interpolates the combined data of the facet. The *blend(ing) functions*, i.e. the weights of the convex combination, limit the influence of the pieces to those edges where they match the data. Compared to the single patch approach, this *buys simpler pieces at the cost of more pieces*. Since surfaces generated by the blending approach typically interpolate a transversal derivative along with the mesh curves, the blend functions play an additional role: they introduce a discontinuity in the second derivative at the data point,

so that the mixed derivatives need not agree. A typical blend function is $b_i b_j/(b_1 b_2 + b_2 b_3 + b_3 b_1)$ where b_1, b_2 and b_3 are the barycentric coordinates of a domain triangle (cf. [Nielson '86]). Discontinuities in higher-order data can be introduced by taking powers of the products $b_l b_m$. This means, however, that consistency of lower-order data has to be enforced by some other means (cf. [Hagen, Pottmann '89]). The degree of blended interpolants is difficult to capture. The table lists, in the 'degree' column, both the degree of the data polynomials, q, and the degree of the (possibly rational) partial interpolant (separated by a '+'), and, in the 'weights' column, the degree of the blend functions (in brackets) and the degree of the rational parameters in terms of the original coordinates (in parentheses).

Only the third, the **splitting approach**, alters the original mesh of neighborhood relationships. Following the example of the Clough-Tocher split, k non-overlapping pieces are used to cover a k-sided mesh facet. The object is to *reduce the degree of the interpolant at the cost of more pieces*. Even though the splitting approach creates points with an even number of neighbors, it can interpolate (original) boundary curves. This is possible because the additional boundaries created by splitting create discontinuities in the higher derivatives just like the rational blend functions above.

Fixing the weight functions *a priori* as in [Goodman '89] or [Höllig '89] is yet another way of removing the nonlinearity in (2.E). This approach and approaches based on an implicit representation ([Sederberg '85], [Bajaj,Ihmsung '89], [Dahmen '89]) have not been included in the table. The reader should be wary that smooth surfaces constructed by any of the algorithms listed in the table can be a far cry from the taut, convex surface its designer may have in mind; the degree of the interpolants is only a first indicator.

Chapter Five

Algorithms for interpolation by C^1 surfaces

This chapter reviews the ideas underlying the algorithms in [Peters '88a '88b '89a '89b], lists shortcomings and restrictions in the original algorithms and gives fixes. To unify the statement of the algorithms and to be specific, some simple functions are defined below.

• is the vector product, × the cross product.

nbrs(k) – returns the number of neighbors of point k.

nbr(k,i) – returns the i^{th} neighbor of point k.

sds(k,i) – returns the number of edges of the i^{th} facet attached to point k.

sds(f) – returns the number of edges of the facet f.

avg($p1, p2, p3, i$) – returns the vector $p1 + \frac{1}{\lambda}(p2 + p3 - 2p1)$, where $\lambda := 2(1 - \cos(\frac{2\pi}{i}))$.

cs($v1, v2$) – returns the scalar $(v1 \bullet v2)/(v1 \bullet v1)$. (Almost, but not quite $\cos(v1, v2)$).

tanproj(v, n) – returns $(v - v \bullet n)n$, the vector component of v perpendicular to n. This is used to project a direction into the tangent plane.

Only one postscript image is displayed in this section since the graphics workstation and corresponding software were unavailable during the last seven months of this thesis work. More surfaces are shown in [Peters '88a '88b '89a '89b].

5.1 Algorithm I: Linearly varying normal along patch boundaries

Idea By prescribing the normal direction, $n(u)$, along the patch boundaries the constraint (2.e) of Lemma 2.2(ii) becomes the local linear set of constraints

$$n(u)p_u(u,0) = 0, \qquad (2.e_u)$$

$$n(u)p_v(u,0) = 0, \qquad (2.e_v)$$

$$n(u)q_w(u,0) = 0, \qquad (2.e_w)$$

for $u \in [0..1]$. If points P_i and their normals N_i are given such that

$$\omega := \frac{N^0(P^0 - P^1)}{N^1(P^1 - P^0)} \geq 0 \qquad (5.1)$$

then the boundary curve connecting two points needs not to have a point of inflection and n can be chosen as a linear interpolant to the normal directions at the points [Peters '88a]:

$$n(u) := [N^0, \omega N^1].$$

Problem To enforce $(2.e_u)$, $(2.e_v)$ and $(2.e_w)$ with [bi]cubic patches, it is sufficient to solve the system (N) of equations at each data point and then compute the interior BB-coefficient(s).

$$\begin{bmatrix} n & 0 & 0 & \dots & 0 \\ n_1 & 0 & 0 & \dots & 0 \\ n_2 & -n_1 & 0 & \dots & 0 \\ 0 & n & 0 & \dots & 0 \\ 0 & n_2 & 0 & \dots & 0 \\ 0 & n_3 & -n_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -n_{k-1} \\ 0 & 0 & 0 & \dots & n \\ 0 & 0 & 0 & \dots & n_k \\ -n_k & 0 & 0 & \dots & n_1 \end{bmatrix} \begin{bmatrix} u_1^0 \\ \vdots \\ u_k^0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2}{3}(P^1 - P)(n + 2n^1) \\ 0 \\ 0 \\ \frac{2}{3}(P^2 - P)(n + 2n^2) \\ 0 \\ \vdots \\ 0 \\ 0 \\ \frac{2}{3}(P^k - P)(n + 2n^k) \\ 0 \end{bmatrix} \tag{5.2}$$

Here $n := n(P)$, $n_i := n(P_i)$, i.e. $[n, n_i, n_{i+1}]$ is a 3×3 submatrix on the diagonal and the u_i^0 are the first BB difference vectors. Solving the system is problematic since its coefficient matrix is rank-deficient at even-points.

A fix The problem is an instance of the vertex enclosure problem (and disappears if the data are 'second-order consistent' [Peters '88a Theorem 5.1]). It is here solved by using parametrizations that are *singular* in the patch corners (see Corollary 3.14) and otherwise deviate least from cubic patches. That is,

$$P_i = B_{ij}^1 = B_{ik}^1 = C_i.$$

The appropriate degree of the patches turns out to be

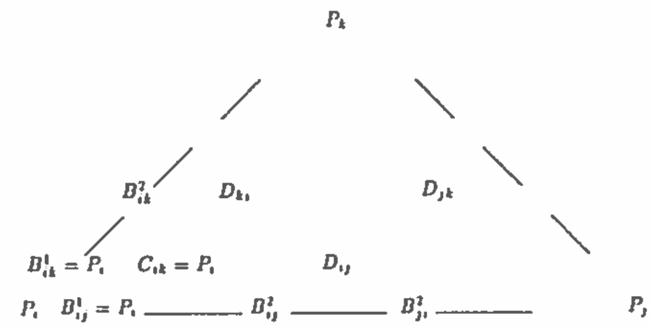
quintic if the facet has 3 edges, or

biquartic if the facet has 4 edges.

Algorithm 5.1

Input A mesh of data points and their normals. The facets all have 3 or 4 edges.

Output A quintic or biquartic C^1 surface that interpolates the mesh.



(5.3) Figure: BB-coefficients for a singular quintic parametrization.

for i = 1:points
  for m = 1:nbrs(i)
    l =nbr(i, m); [e.g. $l = j$ or $l = k$]
    $B_{il}^1 \leftarrow P_i$; $C_{il} \leftarrow P_i$; [singularity at P_i]
    [construct the remaining boundary]
    $\Delta = P_l - P_i$; $\sigma = N_l * \Delta$;
    if quintic **then**
      $t \leftarrow$ tanproj(Δ, N_i); [a vector in the tangent plane]
      $\alpha \leftarrow 3\sigma/(5t * N_l)$;
      if $\alpha \leq 0$
        error [no linear n possible];
      $B_{il}^2 \leftarrow P_i + \alpha t$; [force the second difference into the tangent plane]
    else [biquartic patch]
      $m \leftarrow N^0 \times N^1$; $t \leftarrow m \times N^0$;
      $\alpha \leftarrow \sigma/(m * m)$;
      $B_{il} \leftarrow P_i + \alpha t$;

```
for f = 1:faces        [construct the D_il]
  for i = 1 : sds(f)
    m_i ← N_i × N_{i+1}
    x_i ← P_i + σ(m_i × N_i)/(||m_i||^2)
  solve the sds(f) × sds(f) system
```

$$2m_i \cdot m_i\gamma_i - m_i \cdot m_{i-1}\gamma_{i-1} - m_i \cdot m_{i+1}\gamma_{i+1}$$
$$= (2x_i - x_{i-1} - x_{i+1}) \cdot m_i \qquad (5.4)$$

```
  for i = 1 : sds(f)
    D_i ← x_i + γ_i m_i;
  if sds(f) == 4
    biquartic center coefficient ← average of the surrounding coefficients
```

Correctness of the algorithm We first consider a quintic triangular patch. Since

$$np_u \sim [n^0, n^1][u^0, 4u^1, 6u^2, 4u^3, u^4]$$
$$=:[N^0, \omega N^1][B_{ij}^1 - P_i, 4(B_{ij}^2 - B_{ij}^1), 6(B_{ji}^2 - B_{ij}^2), 4(B_{ji}^1 - B_{ji}^2), P_j - B_{ji}^1]$$

is a polynomial of degree five and the setup is symmetric, it suffices to show that the first three coefficients of np_u,

$$n^0u^0, \quad n^1u^0 + 4n^0u^1, \quad 4n^1u^1 + 6n^0u^2,$$

are zero. By choosing $B_{il}^1 = P_i$ and forcing B_{il}^2 to lie in the tangent plane, e.g. $B_{il}^2 = P_i + \alpha t$, where t is the projection of $P_l - P_i$ into the tangent plane at P_i, the first two coefficients vanish and the boundary curve is indeed perpendicular to the normal. It remains to show that

$$6u^2n^0 + 4u^1n^1 = 0 \quad \text{and} \quad 4u^3n^0 + 6u^2n^1 = 0.$$

Since $u^2 = (P^1 - P^0) - u^1 - u^3 =: \Delta - u^1 - u^0$, we obtain

$$6u^2n^0\bar{u} + 4u^1n^1 = 6\Delta n^0 - 6u^3n^0 + 4u^1n^1$$
$$6u^2n^1\bar{u} + 4u^3n^0 = 6\Delta n^1 - 6u^1n^1 + 4u^3n^0$$

or, equivalently,

$$5u^1n^1 = 3\Delta(2n^0 + 3n^1)$$
$$5u^3n^0 = 3\Delta(3n^0 + 2n^1).$$

Hence, the third coefficient vanishes if

$$\alpha = \frac{3\Delta(2n^0 + 3n^1)}{5tn^1} = \frac{6\Delta N^0/\omega + 9\Delta N^1}{5tN^1} = \frac{-6\Delta N^1 + 9\Delta N^1}{5tN^1} = \frac{3\sigma}{5tN^1}.$$

The analysis of the biquartic case is simpler. Again, choosing $B_{ij}^1 = P_i$ sets to zero the first and the last entry in

$$np_u \sim [n^0, n^1][u^0, 3u^1, 3u^2, u^3]$$
$$=:[N^0, \omega N^1][B_{ij}^1 - P_i, 3(B_{ij} - B_{ij}^1), 3(B_{ji}^1 - B_{ij}), P_j - B_{ji}^1].$$

Forcing the middle coefficient of the boundary, B_{ij}, to lie on the intersection of the tangent planes at P_i and P_j sets to zero the second and fourth entry:

$$n^0(B_{ij} - B_{ij}^1) = 0 = n^1(B_{ji}^1 - B_{ij}).$$

The remaining term is zero by choice of ω:

$$n^0(B_{ji}^1 - B_{ij}) + n^1(B_{ij} - B_{ij}^1) = (n^0 + n^1)(B_{ji}^1 - B_{ij}^1) = (N^0 + \omega N^1)\Delta = 0.$$

The coefficient B_{ij} is underdetermined, since it needs only to satisfy

$$N^0(B_{ij} - P_i) = 0 \quad \text{and} \quad N^1(B_{ij} - P_j) = 0.$$

The above version of the algorithm adds the *ad hoc* additional constraint

$$(N^0 \times N^1)(B_{ij} - P_i) = 0.$$

By setting $B_{i,i+1} = P_i$ and $C_{il} = P_i$, the first two coefficients of np_* vanish. Noting that, in both the quintic and the biquartic case, np_* is a quintic univariate polynomial, it suffices to enforce for each boundary in the quintic case

$$(D_{il} - B^2_{il})N_i = 0 \text{ and } (D_{il} - B^2_{li})N_l = 0, \tag{5.5}$$

respectively in the biquartic case

$$(D_{il} - B_{il})N_i = 0 \text{ and } (D_{il} - B_{li})N_l = 0,$$

where D_{il} is the middle coefficient of the first off-boundary layer of coefficients. Since the problem is underdetermined, one can e.g. minimize the deviation of the quintic from a cubic patch. That is, since a cubic patch has just one interior coefficient, one can minimize the distance of the three D_{il} from each other subject to the $3 * 2$ constraints of type (5.5). Then (D_{ij}, D_{jk}, D_{ki}) are the solution of

$$\min \frac{1}{2}\left(\|D_{ij} - D_{jk}\|^2 + \|D_{jk} - D_{ki}\|^2 + \|D_{ki} - D_{ij}\|^2\right) \qquad \text{s.t.} \quad n_i D_{ij} = n_i B^2_{ij}, \quad i,j \in \{1,2,3\}, i \neq j. \tag{5.6}$$

One can avoid calling a minimization subroutine by first computing a point x_{ij} on the tangent planes of P_i and P_j and then minimizing in the direction m_{ij} perpendicular to N_i and N_j. The solution of

$$N_i(x_{ij} - P_i) = 0$$
$$N_j(x_{ij} - P_j) = 0$$
$$(N_i \times N_j)(x_{ij} - P_i) = 0$$

yields a suitable x_{ij}, namely

$$x_{ij} := P_i + \frac{N_j(P_j - P_i)}{\|m_{ij}\|^2}(m_{ij} \times N_i), \quad \text{where } m_{ij} := N_i \times N_j.$$

Hence

$$D_{ij} = x_{ij} + \gamma_{ij} m_{ij}$$

for scalars γ_{ij} that minimize

$$\min \sum_{a,b \in \{ij,jk,ki\}} \frac{1}{2}\|(x_a - x_b) + (\gamma_a m_a - \gamma_b m_b)\|^2. \tag{5.6'}$$

Since (5.6') is an unconstrained minimization problem, the γ_{ij} can be determined explicitly as the solution of the diagonally dominant system (5.4) that arises when setting to zero the derivative of the function to be minimized. The derivation and formulas for minimizing the deviation of a biquartic from a biquadratic patch are analogous.

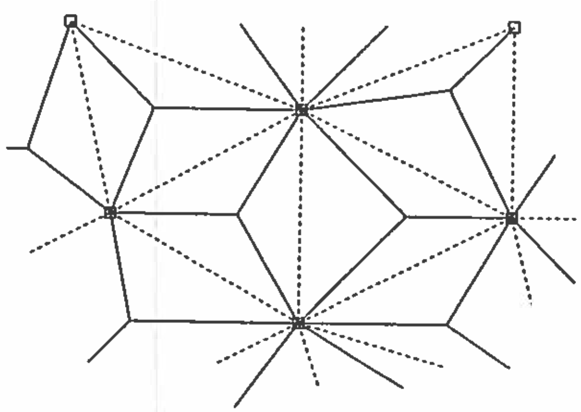
Since the second ring of coefficients around a vertex falls into the tangent plane while the first ring coincides with the data point, the tangent plane of the surface is welldefined at the vertex. For display, the normal at the patch vertices has to be provided explicitly, e.g. from the input data, since it is not available from the first differences of the parametrization.

5.2 Algorithm II: Quadrangulation with diagonal fitting

Idea By splitting each triangle into three subtriangles and combining any two subtriangles across an original edge, one can convert a triangular pattern into a pattern of quadrilaterals (see Figure 5.7). The surface construction obtained by solving (2.E) for bicubic patches with weight functions

$$\lambda = [\lambda^0, 1], \quad \mu = \nu = 1,$$

even allows for a free choice of the mixed derivatives at the original vertices.



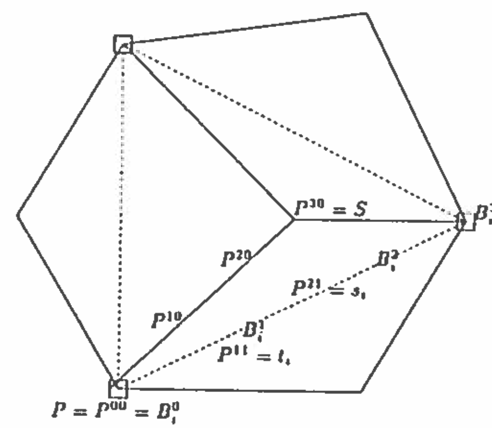
(5.7) Figure: From a triangulation to a quadrangulation. A box indicates a data point, no box a splitting point.

Problem None of the twist estimates tried in [Peters 89d] effectively controls 'buckling' of the patches. Additionally, one wants to find a good default for the length of tangent vectors at the data points.

A fix The free parameters can be fixed so as to minimize the distance between the diagonal $u = v$ of the tensor product patch and some desirable cubic curve

$$b := [B^0, 3B^1, 3B^2, B^3]$$

that connects the data points B^0 and B^3.



(5.8) Figure: Notation for Algorithm 5.II.

Algorithm 5.II

Input A mesh of cubic curves $b := [B^0, 3B^1, 3B^2, B^3]$.

Output A piecewise bicubic C^1 surface (cf. Figure 5.8) that interpolates the cubic curve mesh at the data points and approximates it otherwise.

step 0 Set $\theta := 2\pi/n$, $c := 2\cos\theta$, $\gamma := 1/(2\cos(\theta/2))$ and $C := 2 + c$.

step 1 Set $P^{00} = P$ and

$$P_l^{01} = P + 2\gamma \sum_{k=1}^{n} B_k^1 \cos((l-k+1/2)\theta).$$

step 2 Set $3x_i := c(B_i^2 + 3B_i^1 + P) - 2P + (7-c)(P_i^{01} + P_{i+1}^{01})$. Determine the twist $t_i := P_i^{11}$ from the $n \times n$ system of equations

$$t_{i-2} + 2Ct_{i-1} + (2+C^2)t_i + 2Ct_{i+1} + t_{i+2} = x_{i-1} + Cx_i + x_{i+1}.$$

Set

$$P_i^{02} := P_i^{01} + \frac{1}{c}(-u_i^0 + 3(t_{i-1} - P_i^{01} + t_i - P_i^{01}))$$

step 3 Set $S := P_l^{30} = (b_1 + b_2 + b_3)/3$ and

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 6b_2 + 2cu_2^1 + 2u_2^2 \\ 6b_3 + 2cu_3^1 + 2u_3^2 \\ 6b_1 + 2cu_1^1 + 2u_1^2 \end{bmatrix}.$$

Correctness of the algorithm The coefficients of the $u = v$ section of a patch

$$p(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} P^{ij}(1-u)^i u^{3-i}(1-v)^j v^{3-j}$$

are

$$[P^{00}, 3(P^{01}+P^{10}), 3(P^{02}+3P^{11}+P^{20}), P^{03}+9P^{12}+9P^{21}+P^{30},$$

$$3(P^{13}+3P^{22}+P^{31}), 3(P^{23}+P^{32}), P^{33}].$$

This is to be compared with the degree-raised cubic

$$b = [B^0, 3(B^1+B^0), 3(B^2+3B^1+B^0), B^3+9B^2+9B^1+B^0,$$

$$3(B^3+3B^2+B^1), 3(B^3+B^2), B^3].$$

To enforce (2.E) with the choice

$$\lambda^0 = 2c := 2\cos\theta, \quad \text{where } \theta := \frac{2\pi}{n}$$

and n is the number of neighbors at the data point $P := P^{00} = B^0$, the coefficients of

$$\lambda p_u - \mu p_v - \nu q_w =$$

$$[2c, 1][u^0, 2u^1, u^2] - [v^0, 3v^1, 3v^2, v^3] - [w^0, 3w^1, 3w^2, w^3]$$

are set to zero. To set the first coefficient to zero, the tangents $u_i^0 := P_i^{01} - P$ at P are determined so as to minimize the distance between the diagonal $u = v$ of the tensor product patch

$$(P^{01} + P^{10})/2 - P = u_i^0 + u_{i+1}^0$$

and the tangent vector

$$(B^1 + B^0)/2 - P = B_i^1 - P$$

of given cubic $b := [B^0, 3B^1, 3B^2, B^3]$. That is, one solves the quadratic least squares problem

$$\min \sum_{i=1}^{n} \|u_i^0 + u_{i+1}^0 - (B_i^1 - P)\|^2$$

$$\text{s.t.} \quad u_{i-1}^0 - 2cu_i^0 + u_{i+1}^0 = 0 \quad \text{for } i = 1,\ldots,n \qquad (5.9)$$

(Since the normal does not explicitly appear in the constraints, the minimization can change the tangent plane.) The nullspace of the circulant constraint matrix of each coordinate is spanned by

$$(\omega^k)_{k=1,\ldots,n} \text{ and } (\omega^{-k})_{k=1,\ldots,n},$$

where ω is the n^{th} root of unity. Hence

$$u_k^0 = A\omega^k + B\omega^{-k} \quad \text{for some } A, B \in \mathbb{R}^3$$

and (5.9) is equivalent to

$$\min_{A,B} \sum_{k=1}^{n} \|2\cos(\frac{\theta}{2})(A\omega^{k+1/2} + B\omega^{-(k+1/2)}) - (B_k^1 - P)\|^2$$

which has the same solution as the 6×6 system

$$\sum_{k=1}^{n} \omega^{2k+1} A + nB = \gamma \sum_{k=1}^{n} (B_k^1 - P)\omega^{k+1/2}$$

$$\sum_{k=1}^{n} \omega^{-(2k+1)} B + nA = \gamma \sum_{k=1}^{n} (B_k^1 - P)\omega^{-(k+1/2)}$$

where $\gamma := 1/(2\cos(\theta/2))$. Since $\sum_{k=1}^{n} \omega^{2k+1} = 0$ and $\sum_{k=1}^{n} \omega^{(k+1/2)} = 0$, we can read off

$$A = \gamma \sum_{k=1}^{n} B_k^1 \omega^{-(k+1/2)} \quad B = \gamma \sum_{k=1}^{n} B_k^1 \omega^{k+1/2}$$

and hence

$$u_l^0 = \gamma(\sum_{k=1}^{n} B_k^1 \omega^{-(k+1/2)+l} + \sum_{k=1}^{n} B_k^1 \omega^{k+1/2-l})$$

$$= 2\gamma \sum_{k=1}^{n} B_k^1 \cos((l - k + 1/2)\theta).$$

The second coefficient of $\lambda p_u - p_v - q_w$ vanishes if

$$P_i^{02} = P_i^{01} + \frac{1}{c}(-u_i^0 + 3(t_{i-1} - P_i^{01} + t_i - P_i^{01}))$$

for $t_i := P_i^{11}$. The t_i are chosen to minimize

$$\sum_{i=1}^{n} \|(P_i^{02} + 3t_i + P_i^{20}) - (B_i^2 + 3B_i^1 + P)\|^2$$

or, equivalently, as the solution of

$$\min_{t_i} \sum_{i=1}^{n} \|t_{i-1} + (2+c)t_i + t_{i+1} - x_i\|^2, \tag{5.10}$$

where $3x_i := c(B_i^2 + 3B_i^1 + P) - 2P + (7-c)(P_i^{01} + P_{i+1}^{01})$. The system

$$t_{i-2} + 2Ct_{i-1} + (2 + C^2)t_i + 2Ct_{i+1} + t_{i+2} = x_{i-1} + Cx_i + x_{i+1}$$

with $C := 2+c$ has the same solution as (5.10). It is not diagonally dominant and may have to be solved by least squares.

The fourth coefficient of $\lambda p_u - p_v - q_w$ vanishes if the splitting point is chosen as the average of the surrounding coefficients P^{02}, since then

$$u^2 = v^3 + w^3.$$

With the splitting point fixed, the third coefficient of $\lambda p_u - p_v - q_w$ is forced to zero by computing the twists coefficients s_1, s_2, and s_3 at the splitting point from

$$2cu^2 + 2u^1 = 3(s_i - P_i^{02} + s_{i+1} - P_i^{02}).$$

Conclusion An obvious extension of the approach is to allow initial tesselations with n-sided facets. The facets are split into triangles and then recombined leading to n-points as splitting vertices.

5.3 Algorithm III: Splitting-and-Averaging

Idea Covering an n-sided domain by n non-overlapping patches, i.e. splitting the domain into n triangles, and computing the additional boundary curves by averaging, solves simultaneously the vertex enclosure problem and generates enough degrees of freedom to do so with a low degree (total degree cubic!) interpolant (cf. Corollary 3.19). In fact, *all* coefficients of the interpolating surface can be computed as averages of the data. As a side-effect, the interpolating surface is parametrically smooth across the splitting curves. A full analysis of the algorithm and its implementation for arbitrary topologies can be found in [Peters 88a/90a]. The following describes only the generic construction for 3- and 4-sided facets and quadratic boundary curves.

Algorithm III Given a mesh of cubic curves, SplAv (for Splitting and Averaging) determines the coefficients from the mesh curves inward towards the splitting point. That is (see Figure 5.11), with the coefficients labeled P and D given, the algorithm first determines all coefficients A, then Q, then B and finally M. N^k is the normal at P^k.

Algorithm SplAv [special case: quadratic boundaries]

Input A mesh of quadratic curves (with derivative $\tilde{u} := [u^0, u^1]$) with 3- or 4-sided facets.

Output A cubic C^1 surface that interpolates the mesh.

$A_i^k \leftarrow \text{avg}(P^k, D_{i-1}^k, D^k, \text{sds}(k,i));$
$u^0 \leftarrow D_i^k - P^k;\ u^1 \leftarrow P^l - D_{j-1}^l;$
$v^0 \leftarrow A_i^k - P^k;\ v^1 \leftarrow A_j^l - D_{j-1}^l;$
$u_\perp^0 \leftarrow N^0 \times u^0;\ u_\perp^1 \leftarrow N^1 \times u^1;$

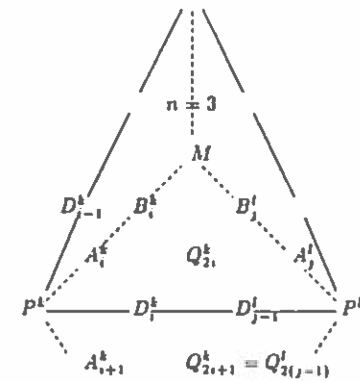
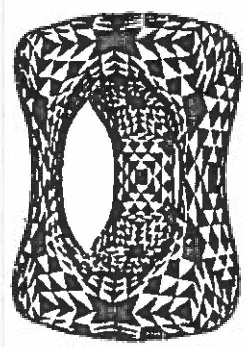


Figure 5.11: Notation for the cubic patches with quadratic boundary.

$Q_{2i}^k \leftarrow D_i^k + (\text{cs}(u_\perp^0, v^0)u_\perp^1 + \text{cs}(u_\perp^1, v^1)u_\perp^0 + \text{cs}(u^0, v^0)u^1 + \text{cs}(u^1, v^1)u^0)/2;$
$B_i^k \leftarrow \text{avg}(A_i^k, Q_{2i-1}^k, Q_{2i}^k, \text{sds}(k,i));$
$M \leftarrow \text{avg}(B_1^k, B_0^k, B_2^k, \text{sds}(f));$

The resulting surface has no cusps along the curve mesh.



(5.12) **Figure:** Three semi-doughnuts meet at an angle of $\frac{2}{3}\pi$.

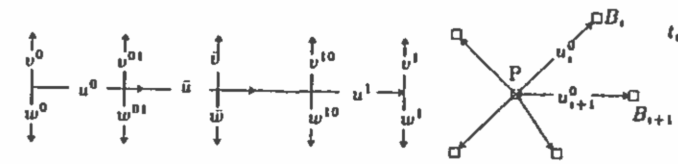
5.4 Algorithm IV: One biquartic per facet

Certain geometric input data allow for simpler, lower degree interpolants than is possible in general. For example, a cubic 2-direction mesh with a welldefined normal at the mesh points can be interpolated and smoothly covered by bicubic patches [Gordon '69]. Underlying is the fact that some data allow for particularly simple weight functions (connecting-maps) in the surface construction, e.g. $\lambda = 0$, $\mu = \nu = 1$ in the above case.

Problem A general surface fitting method must work for general data and must therefore provide high-degree weight functions and patches. However, the algorithm should take advantage of special features in the data and apply lower-degree weight functions and patches whenever possible. The transition from generic to special data should be continuous, i.e. the surface should change gradually when the data are perturbed and lose their special features.

It is not obvious that this can always be achieved, since special data often give rise to numerical singularities (see below).

A fix Averaging, in particular degree-raising, is a useful technique to deal with rank deficiencies induced by special geometric data. That is, the generic construction (weight function and patch) will degenerate to the degree-raised solution of the special construction as the data approach the special data. This guarantees continuity of the transition and is illustrated by an algorithm from [Peters '89b] whose aim is to find a single (triangular or rectangular) patch to cover a mesh facet while interpolating cubic boundary curves. In particular, we look at the construction for two abutting tensor-product patches or 4-4 configuration.



(5.13) **Figure:** Difference vectors (left) and BB-coefficients at P (right).

Algorithm 5.IV: Interpolation of a cubic mesh without splitting
special case: 4-4 configuration

The algorithm determines the BB-coefficients proceeding from the data points to the interior.

Input A mesh of cubic curves that allows for a solution of the vertex enclosure constraint and has only 4-sided facets.

Output A biquartic C^1 surface that interpolates the mesh of curves with one patch per facet.

step 1 [Determine the tangent ratios] For each data point, set

$$\lambda_i^0 := \eta_i, \quad \mu_i^0 := k_i \quad \nu_i^0 := 1 - k_i$$

with k_i and η_i defined by $\eta_i u_i^0 = k_i v_i^0 + (1-k_i) w_i^0$.

step 2 [Determine the twists] For each data point P, set $n :=$ nbrs(P), $\lambda^* := (2\lambda(0) + \lambda(1))/3$, and assemble the matrix

$$K := \begin{bmatrix} k_1 & 1-k_1 & \dots & 0 & 0 \\ 0 & k_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \dots & k_{n-1} & 1-k_{n-1} \\ 1-k_n & 0 & \dots & 0 & k_n \end{bmatrix}$$

and the vector

$$\begin{cases} R := (3\lambda_i^0 u_i^1 + 3\lambda_i^1 u_i^0 + 4B_i - \mu_i^1 v_i^0 - \nu_i^1 w_i^0)_{i=\{1,\dots,n\}} & n \text{ odd} \\ R := (3\lambda_i^0 u_i^1 + 4B_i - \mu_i^1 v_i^0 - \nu_i^1 w_i^0)_{i=\{1,\dots,n\}} & n \text{ even} \end{cases}$$

If the number of incident edges is odd, set $\lambda_i^1 := \lambda^*$ and solve

$$KT(k) = R(k)$$

for each coordinate $k \in \{1,2,3\}$. If the number of incident edges is even, the constraint system is underdetermined. Choose t^* as a desirable choice of twist vector and solve the quadratic least squares problem

$$\min_{T,L} \frac{1}{2} \sum_{i=1}^{n} (\|t_i^* - t_i\|^2 + \frac{1}{|k^0 - k^1|} \|\lambda_i^* - \lambda_i^1\|^2)$$

$$\text{s.t.} \begin{bmatrix} K & 0 & 0 & U(x) \\ 0 & K & 0 & U(y) \\ 0 & 0 & K & U(z) \end{bmatrix} \begin{bmatrix} T(x) \\ T(y) \\ T(z) \\ L \end{bmatrix} = \begin{bmatrix} R(x) \\ R(y) \\ R(z) \end{bmatrix}, \tag{5.14}$$

where $L := (\lambda_i^1)_{i=1\dots n}$, $T := (t_i)_{i=1\dots n}$ and

$$U(x) := -3 \begin{bmatrix} u^0(x)_1 & 0 & \dots & 0 \\ 0 & u^0(x)_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u^0(x)_n \end{bmatrix}.$$

step 3 [Determine the interior coefficients] Let k^0 be the weight, determined in step 1, corresponding to the left end point of the curve segment and k^1 the one corresponding to the right end point. If $|k^0 - k^1| >$TOL, compute the middle coefficients of the first layer from

$$\begin{aligned} \mu^0 \bar{v} + (1-\mu^0)\bar{w} &= \frac{1}{6}(3\lambda^{10} u^0 + 6\lambda^{01} \bar{u} + \lambda^0 u^1 - 4\mu^1 v^{01} - 4\nu^1 w^{01}) \\ \mu^1 \bar{v} + (1-\mu^1)\bar{w} &= \frac{1}{6}(3\lambda^{01} u^1 + 6\lambda^{10} \bar{u} + \lambda^1 u^0 - 4\mu^0 v^{10} - 4\nu^0 w^{10}). \end{aligned} \tag{5.15}$$

Otherwise set

$$6v^* := 4v^{01} + 4v^{10} - (v^0 + v^1), \quad 6w^* := 4w^{01} + 4w^{10} - (w^0 + w^1)$$

and solve the quadratic least squares problem

$$\min \quad \frac{1}{2}(\|v^* - \bar{v}\|^2 + \|w^* - \bar{w}\|^2)$$

$$\text{s.t. } \lambda^1 u^0 + 4\hat{\lambda}\bar{u} + \lambda^0 u^1 = \mu\bar{v} + \nu\bar{w}. \tag{5.16}$$

The central coefficient is free and can be chosen, e.g. as an average of the surrounding coefficients.

Discussion We focus on step 3 of the algorithm in which the degree-raising principle is used twice.

(5.17) Theorem. *As $k^0 \to k^1$, the surface construction (weight function and patch) converges to the (degree-raised) construction of $k^0 = k^1$ if (5.14) can be solved for $\lambda_i^1 = \lambda^*$. Otherwise the deviation from the degree-raised construction is minimal.*

Proof. Consider the constraints $\lambda\tilde{u} = \mu\tilde{v} + \nu\tilde{w}$ where

$$\tilde{u} \sim [u^0, 2\bar{u}, u^1]$$
$$\tilde{v} \sim [v^0, 4v^{01}, 6\bar{v}, 4v^{10}, v^1]$$
$$\tilde{w} \sim [w^0, 4w^{01}, 6\bar{w}, 4w^{10}, w^1].$$

Since the ratios $k^0/(1-k^0)$ and $k^1/(1-k^1)$ computed in step 1 of the algorithm are not necessarily equal, μ and ν must in general be at least linear. The choice $\lambda \sim [\lambda^0, 3\lambda^{01}, 3\lambda^{10}, \lambda^1]$, $\mu \sim [\mu^0, \mu^1]$, $\nu \sim [\nu^0, \nu^1]$ leads to the C^1 constraints

$$\lambda^0 u^0 = \mu^0 v^0 + \nu^0 w^0 \quad (2.E_0)$$
$$3\lambda^{01}u^0 + 2\lambda^0\bar{u} = 4\mu^0 v^{01} + \mu^1 v^0 + 4\nu^0 w^{01} + \nu^1 w^0 \quad (2.E_1)$$
$$3\lambda^{10}u^0 + 6\lambda^{01}\bar{u} + \lambda^0 u^1 = 4\mu^1 v^{01} + 6\mu^0\bar{v} + 4\nu^1 w^{01} + 6\nu^0\bar{w} \quad (2.E_2)$$
$$3\lambda^{01}u^1 + 6\lambda^{10}\bar{u} + \lambda^1 u^0 = 4\mu^0 v^{10} + 6\mu^1\bar{v} + 4\nu^0 w^{10} + 6\nu^1\bar{w}$$
$$3\lambda^{10}u^1 + 2\lambda^1\bar{u} = 4\mu^1 v^{10} + \mu^0 v^1 + 4\nu^1 w^{10} + \nu^0 w^1$$
$$\lambda^1 u^1 = \mu^1 v^1 + \nu^1 w^1$$

Step 1 and step 2 of the algorithm enforce all constraints of type $(2.E_0)$ and $(2.E_1)$ and thereby determine all coefficients, except for $\bar{v}$ and $\bar{w}$ which are computed from the constraints (5.15). If $k^0 \neq k^1$, i.e. in the generic case, one can solve for $\bar{v}$ and $\bar{w}$. If, however, $k^0 = k^1$, (5.15) is only solvable if the right-hand sides agree. The freedom in the choice of λ^{01} and λ^{10} gives hope that this can be achieved. If $k^0 = k^1$, the simplest choice of weight functions is $\lambda \sim [\lambda^0, 2\bar{\lambda}, \lambda^1]$, $\mu \sim [\mu]$ and $\nu \sim [\nu]$. This leads to the constraints

$$\lambda^0 u^0 = \mu v^0 + \nu w^0 \quad (2.E_0)$$
$$2\bar{\lambda}u^0 + 2\lambda^0\bar{u} = 4\mu v^{01} + 4\nu w^{01} \quad (2.E_1')$$
$$\lambda^1 u^0 + 4\bar{\lambda}\bar{u} + \lambda^0 u^1 = \mu\bar{v} + \nu\bar{w} \quad (2.E_2')$$
$$2\bar{\lambda}u^0 + 2\lambda^0\bar{u} = 4\mu v^{01} + 4\nu w^{01}$$
$$\lambda^1 u^1 = \mu v^1 + \nu w^1$$

Again, step 1 and step 2 of the algorithm enforce all constraints of type $(2.E_0)$ and $(2.E_1)$, and $\bar{v}$ and $\bar{w}$ are computed in step 3. Raising the degree of the weight functions in (2.E′) by one gives

$$\lambda \sim [1,1][\lambda^0, 2\bar{\lambda}, \lambda^1] = [\lambda^0, 2\bar{\lambda} + \lambda^0, 2\bar{\lambda} + \lambda^1, \lambda^1],$$

$$\mu \sim [1,1][\mu] = [\mu, \mu] \quad \text{and} \quad \nu \sim [1,1][\nu] = [\nu, \nu].$$

The new set of constraints is equivalent to the constraints for the case $k^0 \neq k^1$ exactly if

$$3\lambda^{01} = 2\bar{\lambda} + \lambda^0 \quad \text{and} \quad 3\lambda^{10} = 2\bar{\lambda} + \lambda^1, \quad (5.18)$$

$$\mu^0 = \mu^1 = \mu \quad \text{and} \quad \nu^0 = \nu^1 = \nu. \quad (5.19)$$

Hence, if (5.18) holds in the limit as $k^0 \to k^1$, the construction according to (2.E) degenerates to the construction by solving (2.E′).

In step 2, (5.18) is explicitly enforced if the number of incident curves at the current data point is odd. If the number is even, the deviation from (5.18) is minimized by the choice of λ^*. In particular, if the vertex enclosure problem (5.14) is solvable for $\lambda = \lambda^*$, e.g. when the collinearity constraint (3.6.a) holds, then the transition is smooth. ♣

(5.20) Theorem. *If $k^0 = k^1$ and $\lambda = 0$, then the construction becomes the construction of a degree-raised bicubic patch.*

Proof. If $\lambda = 0$, then (5.16) becomes

$$0 = \mu\tilde{v} + \nu\tilde{w}$$

which, by ($2.E_1$), is satisfied by $\tilde{v} = 4v^{01} + 4v^{10} - (v^0 + v^1)$ and $\tilde{w} = 4w^{01} + 4w^{10} - (w^0 + w^1)$. Raising the degree of the cubic $[a^0, 3a^{01}, 3a^{10}, a^1]$ yields

$$[a^0, a^0 + 3a^{01}, 3a^{01} + 3a^{10}, a^1 + 3a^{10}, a^1].$$

Hence, if

$$a^0 = v^0 \text{ and } a^0 + 3a^{01} = 4v^{01},$$

then v is the middle coefficient of a degree raised cubic iff

$$6v = 6\bar{v}^* = 3a^{01} + 3a^{10} = 4(v^{01} + v^{10}) - (v^0 + v^1).$$

♣

Chapter Six

Higher-order continuity

Since the first and second fundamental forms exhaust the geometric invariants in three dimensions (e.g. [Klingenberg Prop. 3.5.6]), the characterization of higher-order continuity is based on reparametrization and the chain rule. Since [Hahn '89] already gives a detailed exposition of C^k continuity, only the salient points for the purpose of this thesis are developed and extended below. In particular, the following definitions paraphrase [Hahn '89, Definition 2.1 and 3.1].

(6.1) Definition. *Two C^k patches $p_1 : \Omega_1 \to \mathbb{R}^3$ and $p_2 : \Omega_2 \to \mathbb{R}^3$ join with* **C^k continuity** *across the domain edges E_1, E_2 if there exists a C^k connecting-map ϕ with $\phi(E_1) = E_2$ such that*

$$J^k p_{1|E_1(s)} = J^k p_{2|E_2(s)} \circ J^k \phi_{|E_1(s)}$$

for $s \in [0..1]$. Here a C^k **patch** *is a k-times continuously differentiable map $p : \Omega \to \mathbb{R}^3$, whose domain, Ω, is a closed, polygonal subset of $\mathbb{R}^2$ and a C^k* **connecting-map** *is an invertible C^k map that maps interior points of Ω_1 to exterior points of Ω_2. The k-***jet** *of p at X is the sequence of derivatives of order $\le k$ at X: $J^k p_{|X} := (D^j p_{|X})_{j=0,\dots,k}$. The composition of jets $J^k f_{|X}$, $J^k g_{|Y}$ is defined by $J^k g_{|Y} \circ J^k f_{|X} := J^k(g \circ f)_{|X}$, provided $f(X) = Y$. If $\phi = \mathrm{id}$ in the above, then p_1 and p_2 are said to join with* **parametric C^k continuity**.

From Chapter 2, we recall that one has to be careful in restricting the class of connecting-maps when the definition is unsymmetric as it is here, i.e. when only one side of the continuity constraint is a composite map; for example if ϕ is polynomial, one obtains only a sufficient, but not necessary constraint for C^k continuity. Hahn requires additionally that any C^k patch be regular, i.e. that rank $Dp = 2$. While regularity of the patch is not necessary for C^k continuity of the resulting surface, regularity implies that $J^k p_{i|X}$ is left-invertible, i.e. that there exists a $J^k q_{|p_i(X)}$ such that

$$J^k q_{|p_i(X)} \circ J^k p_{i|X} = J^k \mathrm{id}_{|X}.$$

This is the crucial observation for obtaining the following necessary condition at an enclosed vertex.

(6.2) Theorem. [Hahn '89, Theorem 7.1] *Let $(p_i)_{i=1,\ldots,n}$ be regular patches with $p_i(C_i) = Q$ and $J^k p_{i|C_i} = J^k p_{i+1|C_{i+1}} \circ J^k \phi_{i+1,i|C_i}$, where $\phi_{i+1,i}$ are connecting-maps with $\phi_{i+1,i}(C_i) = C_{i+1}$. Then*

$$J^k \phi_{1,n|C_n} \circ J^k \phi_{n,n-1|C_{n-1}} \circ \ldots \circ J^k \phi_{2,1|C_1} = J^k \mathrm{id}_{|C_1}.$$

Proof. By substitution

$$J^k p_{1|C_1} \circ J^k \phi_{1,n|C_n} \circ J^k \phi_{n,n-1|C_{n-1}} \circ \ldots \circ J^k \phi_{2,1|C_1} = J^k p_{1|C_1}.$$

The result follows from the regularity of p_1, the associativity of jet composition and the neutrality of $\mathrm{id}_{|C_1}$. ♣

We apply the theory to the problem of 'filling an n-sided hole within a C^k parametric continuous patch complex with n non-overlapping patches' treated e.g. in [Gregory, Hahn '89] and [Hahn '89a]. Consider Figure 6.3. The patches q_i^j are given and join with parametric C^k continuity. The patches p_i are sought to form a C^k surface with the surrounding complex of patches q_i^j. Any construction has to satisfy the constraints of Theorem 6.2 at L_i, M_i

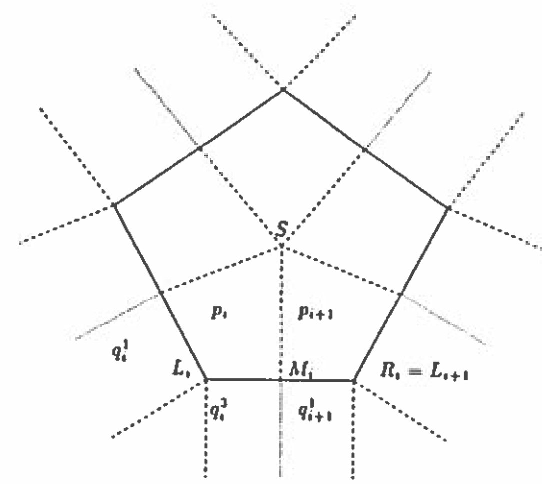


Figure 6.3: Covering a 5-sided hole with 5 'rectangular' patches.

and S. Hahn's construction, [Hahn '89a], uses two *types* of connecting-maps: across any boundary between a given and a new patch

$$\phi^{LM} := \mathrm{id},$$

while between dom p_i and dom p_{i+1}

$$\phi^{SM}(s,t) := ((\mu_i + 1)\alpha(s)t - t, s + \lambda_i \alpha(s)t).$$

Here, the superscript AB indicates that $\phi^{AB}(0)$ is the preimage of A and $\phi^{AB}(1)$ is the preimage of B, α is a C^{2k-1}-function satisfying

$$\alpha(0) = 1, \alpha(1) = 0, \alpha(s) \geq 0 \text{ for } s \in [0,1],$$

$$\alpha'(0) = \ldots = \alpha^{2k-1}(0) = 0, \quad \alpha'(1) = \ldots = \alpha^{k}(1) = 0,$$

$$\lambda_i := \frac{r_{i+1}}{r_i}\left(\cos\theta_i + \cos\theta_{i-1}\frac{\sin\theta_i}{\sin\theta_{i-1}}\right), \mu_i := -\frac{r_{i+1}}{r_i}\frac{\sin\theta_i}{\sin\theta_{i-1}},$$

$r_i := \|D_1 p_i(S)\|$, and θ_i is the angle between $D_1 p_i(S)$ and $D_2 p_{i-1}(S)$. If all r_i are equal, then $\theta_i = \theta := 2\pi/s$. The connecting-map ϕ^{SM} is chosen of so high a degree that the vertex enclosure problems at S and the M_i do not interfere. In fact, the k-jet at S is independent of the given data q_i^j and is obtained by a heuristic; S is connected to the patch complex by (unsymmetric) Hermite interpolation. While this approach avoids the difficulties of vertex enclosure with prescribed data, it unsymmetrically creates a surface of bivariate degree $5k$. In contrast, the following sketches a *symmetric* construction of degree $2k+2$. The connecting-maps are

$$\phi^{LM}(x,y) := \begin{bmatrix} x + \frac{c}{1-c}xy \\ y + \frac{c}{1-c}xy \end{bmatrix} \quad \text{and} \quad \phi^{SM}(x,y) := \begin{bmatrix} y + (1-y)x2c \\ -x \end{bmatrix},$$

where $c := \cos(\theta)$. The idea here is to have a bilinear change of ~~topology~~ geometry from the 4-points, L_i, to the n-point, S.

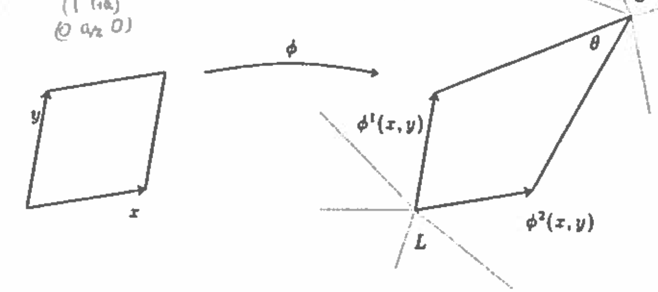
Algorithm 6.V:

Input A complex of rectangular patches joining with k^{th} order parametric continuity and surrounding an n-sided hole.

Output n patches, each of degree $2k+2$ by $2k+2$, that join with the rectangular patch complex and with the neighbor patches to form a C^k surface.

step 1 For each patch p_i (see Figure 6.3), determine the first k transversal derivatives, $(D_2^j p)_{j=1,\ldots,k}$, along the original boundaries as

$$D_2^j p \leftarrow \sum_{i=0}^{j} D_1^{j-i} D_2^i q (ax)^{j-i}(1+ax)^i$$



(6.4) Figure: The bilinear connection-map ϕ.

where $a := c/(1-c)$, $c := \cos(2\pi/s)$, and s the number of edges of the polygonal hole. Raise the degree of $D_2^{k-i}p$ by $i+2$.

Remark Since $\deg(D_2^j p) = \deg(q) + j$, degree-raising is used to give the final surface the uniform degree $2k+2$.

step 2 For each splitting point S, obtain the first $k+1$ derivatives at S by solving the $s(k^2+k)$ system

$$J^k p_{i|S} = J^k(p_{i+1} \circ \phi_i)_{|S}, \quad D_2^{k+1} D_1^n (p_i - p_{i+1} \circ \phi_i))_{|S} = 0, \; n \le k, \quad (6.5)$$

$$\phi_i(x,y) := \begin{bmatrix} y + (1-y)x2c \\ -x \end{bmatrix}$$

in the $s(k+1)^2$ unknowns

$$D_2^n D_1^m p_{i|S}, \quad (n,m) \le (k+1, k+1), i \in \{1, \ldots, s\}.$$

Remark The system is underdetermined. The standard technique is to solve it as the least deviation from a simpler surface with desirable shape properties [Grandine '87]. If s is odd, then a smaller system will do (cf. Lemma 6.7).

Correctness With $\phi := \phi^{LM}$, $p := p_i$ and $q := q_i^1$ or $q := q_{i+1}^1$, we have

$$D_1\phi = \begin{bmatrix} 1+ay \\ ay \end{bmatrix} \quad \text{and} \quad D_2\phi = \begin{bmatrix} ax \\ 1+ax \end{bmatrix}, \tag{6.6}$$

and hence

$$D_2^n p = D_2^n q \circ \phi = D^n q(\underbrace{D_2\phi, \ldots, D_2\phi}_{n \text{ terms}}) = \sum_{i=0}^{n} D_1^{n-i} D_2^i q (ax)^{n-i}(1+ax)^i.$$

An argument that is not detailed here shows that the welldefinedness of the C^k construction together with the validity of Theorem 6.2 at L_i (M_i) imply that there is a unique k-jet at L_i (M_i). This, in turn, implies that the extension across the rim of the polygonal hole is welldefined. The connection-map ψ for the edge L_iM_i and the connection-map φ for the edge $M_{i-1}L_i$ are related by

$$\varphi := G\phi G \quad \text{where} \quad G := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The constraints of Theorem 6.2 are

$$J^k\phi_{|0} \circ J^k\mathrm{id}_{|0} \circ J^k(G\phi^{-1}G)_{|0} = J^k\mathrm{id}_{|0}$$

or, equivalently,

$$J^k(\phi G)_{|0} = GJ^k\phi_{|0}$$

which, in turn, is equivalent to

$$J^k\phi_{1|0} = J^k(\phi_2 G)_{|0}.$$

One checks that $\phi_1(0,0) = 1 = \phi_2(0,0)$,

$$D\phi_1(0,0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = D(\phi_2 G)(0,0), \quad D^2\phi_1(0,0) = \begin{bmatrix} 0 & a \\ 0 & a \end{bmatrix} = D(\phi_2 G)(0,0),$$

and higher derivatives vanish. Similarly, at M_i, the following diagram has to commute

$$\begin{array}{ccc} \mathrm{dom}\, p_i & \xrightarrow{\phi_i^{MS}} & \mathrm{dom}\, p_{i+1} \\ \downarrow \phi_i^{ML} & & \downarrow \phi_{i+1}^{ML} \\ \mathrm{dom}\, q_i & \xrightarrow{G} & \mathrm{dom}\, q_{i+1} \end{array}$$

where p_i and p_{i+1} are patches constructed to fill the polygonal hole, q_i and q_{i+1} are abutting rectangular patches,

$$\phi_i^{ML}(x,y) = \phi_{i+1}^{ML}(x,y) = \phi^{ML}(x,y) := \begin{bmatrix} x+xcy \\ y-cxy \end{bmatrix},$$

$$\phi^{MS}(x,y) := \begin{bmatrix} -x \\ y-y2cx \end{bmatrix} \quad \text{and} \quad G := \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since higher derivatives vanish, it suffices to show that

$$G\phi_i^{ML}{}_{|(0,0)}$$
$$= [0,0]'$$
$$= \phi_{i+1}^{ML} \circ \phi_i^{MS}{}_{|(0,0)}$$

$$D_x G\phi_i^{ML}{}_{|(0,0)}$$
$$= \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$
$$= \mathrm{D}\phi_{i+1}^{ML}(\phi_i^{MS}) D_x \phi_i^{MS}{}_{|(0,0)} = D_x \phi_{i+1}^{ML} \circ \phi_i^{MS}{}_{|(0,0)}$$

$$D_y G\phi_i^{ML}{}_{|(0,0)}$$
$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$= D_y \phi_{i+1}^{ML} \circ \phi_i^{MS}{}_{|(0,0)}$$

$$D_x D_y G\phi_i^{ML}{}_{|(0,0)}$$
$$= \begin{bmatrix} -c \\ -c \end{bmatrix} = \begin{bmatrix} -c \\ c \end{bmatrix} + \begin{bmatrix} 0 \\ -2c \end{bmatrix}$$
$$= \mathrm{D}^2\phi_{i+1}^{ML}(\phi_i^{MS})(D_y\phi_i^{MS}, D_x\phi_i^{MS}) + \mathrm{D}\phi_{i+1}^{ML}(\phi_i^{MS}) D_x D_y \phi_i^{MS}$$
$$= D_x D_y \phi_{i+1}^{ML} \circ \phi_i^{MS}{}_{|(0,0)}$$

Here D_x and D_y are used rather than D_1 and D_2 to avoid ambiguities when differentiating composite maps by stressing that the differentiation is with respect to a direction in the domain of p_i.

To show that **step 2** is also welldefined, one proves the following Claim by Fourier analysis.

(6.7) Claim. [vertex enclosure] *To be able to solve the sk^2 constraints*

$$J^k p_{i|S} = J^k(p_{i+1} \circ \phi_i)_{|S}, \quad \text{for } i \in \{1,\dots,s\} \qquad (6.5)$$

in the sk^2 unknows

$$p_i^{n,m} := D_2^n D_1^m p_{i|S}, \quad (n,m) \le (k,k),\ i \in \{1,\dots,s\},$$

/ it's actually more complicated... → [P9x, CA6L

it is necessary that higher derivatives satisfy $\begin{cases} \lfloor k/2 \rfloor, & \text{if } s \text{ is odd,} \\ k, & \text{if } s \text{ is even and } s \ne 4 \\ 0, & \text{if } s = 4 \end{cases}$

additional constraints. It is sufficient to add either

$$p_i^{k+1,m}, m \le k-1, \quad \text{or} \quad p_i^{n,0}, k+1 \le n \le 2k, \quad \text{for } i \in \{1,\dots,s\} \quad (6.8)$$

as unknowns to be able to solve (6.5).

Remark If the surrounding patch complex can be modified, e.g. by imposing conditions at the rim of the rectangular patch complex, then the algorithm will succeed with patches of degree $2k+1$. Alternatively, one could generalize Gregory's approach and use rational patches that are k^{th}-order smooth but not $k+1^{\text{st}}$-order smooth at S.

Chapter Seven

Outlook

This thesis presented a number of algorithms for fitting smooth surfaces to 3D data. Additional criteria, besides simplicity and smoothness, will decide on their value in practice. In any case, it is useful to establish some basic array of techniques and methods for mathematically welldefined problems before trying to tackle illdefined concepts like 'shape' or 'tautness'. While the engineer may not be interested in C^k-surfaces for $k > 2$, the study of higher order smoothness clarifies the underlying algebraic structure. This is in particular true for the characterization of the k^{th}-order vertex enclosure problem.

The restriction, in Algorithm 6.V, that the surrounding complex be parametrically continuous, can be removed. Once the 'hole-filling' construction is understood, one has actually a general construction where the input consists of a mesh of 'k-jets' and the output is an interpolating surface. This and the proof of Lemma 6.7, currently formulated in the "intermediate level language of (total) derivatives" rather than the "high level language of k-jets" [Hahn '89, Section 6], need to be made precise in a forthcoming report.

References

Andersson, E.,R. Andersson, M. Boman, T. Elmroth, B. Dahlberg, B. Johansson (1988) Automatic construction of surfaces with prescribed shape, *Computer Aided Design* 20,No 6, p. 317-324

Bajaj, C.L., I. Ihmsung (1989), Hermite interpolation using real algebraic surfaces, *Proc. of the 5th ACM Symposium on Computational Geometry* West Germany.

Barnhill, R.E., G. Birkhoff, W.J. Gordon (1973), Smooth Interpolation in Triangles, *J. Approx. Theory*, 8, p 114-128

Becker, Etienne (1986), Smoothing of Shapes Designed with Free Form Surfaces, *Computer Aided Design*18(4):224-232,1986

Bézier, Pierre (1977), *Essai de Définition Numérique des Courbes et des Surfaces Expérimentales*, doctoral dissertation, Université Pierre et Marie Curie, Paris

Bézier, Pierre (1986), *The Mathematical Basis of the UNISURF CAD System* Butterworths, London Boston Durban Singapore Sydney Toronto Wellington

Boehm, W., G. Farin, J. Kahmann (1984), A survey of curve and surface methods in CAGD, *CAGD* 1, p. 43

de Boor, Carl (1987), B-form basics, *Geometric Modeling: Applications and New Trends*, G. Farin ed., SIAM, Philadelphia.

de Boor, Carl, Klaus Höllig, Malcolm Sabin (1987), High accuracy geometric Hermite interpolation, *Computer Aided Geometric Design*, 4: 269-278

do Carmo, M.P. (1976), *Differential geometry of curves and surfaces*, Prentice-Hall, Englewood Cliffs NJ.

de Casteljau, F., (1959), Courbes et surfaces á pôles, ***André Citröen Automobiles***, Paris 1963

Charrot, Peter, John A. Gregory(1984), A pentagonal surface patch for computer aided geometric design, ***Computer Aided Geometric Design***, 1, 87-94

Chiyokura, Hiroaki, Fumihiko Kimura (1983), Design of Solids with Free-form Surfaces, ***Computer Graphics***, 17; No. 3, 289-298

Clough R.W., Tocher J.L. (1965), Finite Element Stiffness Matrices for the Analysis of Plate Bending, ***Proc. 1st Conf. Matrix Methods in Struct. Mechanics*** Wright-Patterson AFB AFFDL TR 66-80 pp 515-545

Coons, S.A.(1967), Surfaces for Computer Aided Design of Space Forms, ***Report MAC-TR-41, Project MAC***, M.I.T.

Degen, W. (1989), Explicit Continuity Conditions for adjacent Bézier Surface Patches, Preprint No.5, Math. Inst. B, Universität Stuttgart

DeRose, Anthony (1985), Geometric Continuity: A Parametrization Independent Measure of Continuity for Computer Aided Design, ***UC Berkeley, California***, thesis

Farin, Gerald (1982), Triangular Bernstein-Bézier patches, ***Computer Graphics and Image Processing*** 20. 272-282

Farin, Gerald (1983), Smooth Interpolation to Scattered 3D-Data, ***Surfaces in CAGD, R.F. Barnhill, W. Boehm, eds.***

Farin, Gerald (1985), A modified Clough-Tocher interpolant, ***Computer Aided Geometric Design*** 2. 19-27

Farin, Gerald (1986), Triangular Bernstein-Bézier patches, ***Computer Aided Geometric Design*** 3.

Farin, Gerald (1988), Curves and Surfaces for Computer Aided Geometric Design, ***Academic Press.***

I.D. Faux and M.J. Pratt (1979), "Computational geometry for design and manufacture", Ellis Horwood

G. Geise (1961), Über berührende Kegelschnitte einer ebenen Kurve, ***ZAMM*** 42 No 7/8, 297-304.

T.N.T. Goodman (1988), Closed surfaces defined from biquadratic splines, ***Report AA 886 University of Dundee***

Gordon, W.J. (1969), Spline-blended surface interpolation through curve networks ***J. Math. Mech.*** 18 No 10, 931-952

Grandine Thomas A. (1987), An Iterative Method for Computing Multivariate C^1 piecewise polynomial Interpolants, ***CAGD*** 4 pp 307-319

Gregory, John A. (1974), Smooth Interpolation without Twist Constraints, ***Computer Aided Geometric Design***, 71-88, R.E. Barnhill, R.F. Riesenfeld, eds. , Academic Press.

Gregory, John A. (1989), Geometric Continuity, ***Mathematical methods in Computer Aided Geometric Design***, T. Lyche and L.L. Schumaker, eds., Academic Press, Boston 1989, 353-371

Gregory, John A. (1990), 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

Gregory, John A., Jörg M. Hahn (1989), A C^2 polygonal surface patch, ***Computer Aided Geometric Design***, 6, 69-75

Hagen, H., Helmut Pottmann (1989), Curvature continuous triangular interpolants, preprint

Hahn, Jörg M. (1987a), Triangular patches within a geometric continuous patch complex, *Technical Report*, TR0787, April 1987

Hahn, Jörg M. (1989), Geometric Continuous Patch Complexes, *CAGD* 6 55-67

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

Jones, A.K.(1988), Nonrectangular surface patches with curvature continuity, *Computer Aided Design*, 20, No. 6: 325-335

Herron, Gary J. (1985), Smooth closed surfaces with discrete triangular interpolants, *CAGD* 2: 297-306

Hirsch, M.W. (1976), *Differential Topology* Graduate Texts in Math. 33, Springer, Berlin

Höllig, Klaus (1986), Geometric Continuity of Spline Curves and Surfaces, *CS Technical Report* 645, UW-Madison, presented at SIGGRAPH '86

Klingenberg, W. (1983) *A course in differential geometry*, Springer Verlag, New York

Liu, Ding-yuan (1986), A Geometric Condition for Smoothness between adjacent Bézier Surface Patches, *Acta Mathematicae Applicatae Sinica* 9, No 4

Liu, Ding-yuan, J. Hoschek (1989), GC^1 Continuity Conditions between adjacent Rectangular and Triangular Bézier surface patches, *Computer Aided Design* 21, No 4, may 1989

Nielson, G.M. (1979), The Side-Vertex method for interpolation in Triangles, *Journal of Approx. Theory* 25, 318-336

Nielson, G.M. (1986), A Transfinite, Visually Continuous, Triangular Interpolant, *Geometric Modeling: Applications and New Trends*, G. Farin ed., SIAM, Philadelphia

Peters, J. (1988a), Local cubic and bicubic C^1 surface interpolation with linearly varying boundary normal, *Computer Aided Geometric Design* 7 1990

Peters, J. (1988b), Local piecewise cubic C^1 surface interpolants via splitting and averaging, CMS Technical Report no. 89-11, U. of Wisconsin-Madison, Sept. 1988, presented at the Oberwolfach meeting April 1989

Peters, J. (1990a), Smooth Mesh Interpolation with Cubic Patches, *Computer Aided Design* 22, No 2, March 1990

Peters, J. (1989a), Local Generalized Hermite Interpolation by Quartic C^2 space curves, *Transactions on Computer Graphics*, July 1989

Peters, J. (1989b), Smooth Interpolation of a Mesh of Curves, to appear in Constructive Approximation, x (1990), xxx-xxx. see also: Local Interpolation of a Cubic Curve Mesh by a Piecewise [Bi]Quartic C^1 Surface without Splitting, CMS Tech. Report No. 89-25 (1989)

Peters, J. (1989c), Local Smooth Surface Interpolation: a Classification, to appear in *Computer Aided Geometric Design*

Peters, J. (1989d), Rectangulation Algorithms: Smooth Surface Interpolation with Bicubics, CMS Tech. Report No. 90-1, U. of Wisconsin-Madison (1989)

Peters, J, M. Sitharam (1989e), Interpolation from C^1 Cubics at the Vertices of an Underlying Triangulation, submitted to SIAM J. of Num. Anal.

Peters, J (1990b), Evaluation of bivariate polynomials in Bernstein-Bézier -form, CMS Tech. Report No. 91-1, U. of Wisconsin-Madison (1990)

Piegl, L. (1988), Hermite-and Coons-like interpolants using rational Bézier approximation form with infinite control points, ***Computer Aided Design*** **20**, No 1, 2–10

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

Powell, M.J.D., M.A. Sabin (1977), Piecewise Quadratic Approximations on Triangles, ***ACM Transactions on Math. Software***, 3 No. 4, Dec. 1977, pp. 316-325

Sabin, M.A. (1968), Conditions for continuity of surface normal between adjacent parametric surfaces, ***Tech. Rep., British Aircraft Corporation Ltd.***

Sabin, M. A. (1977), The use of Piecewise Forms for the numerical representation of shape, ***Computer and Automation Institute, Hungarian Academy of Sciences***, thesis

Sarraga, Ramon F. (1986), G^1 interpolation of generally unrestricted cubic Bézier curves, ***Tech. Report GMR-5424(A)***

Sarraga, Ramon F. (1988), Computer modeling of surfaces with arbitrary shapes, ***Tech. Report GMR-6388***

Sederberg, T. W.(1985), Piecewise algebraic surface patches ***Computer Aided Geometric Design*** **2**, pp 53-59

Shirman, Leon A., Carlo H. Séquin (1987), Local surface interpolation with Bézier patches, ***Computer Aided Geometric Design*** 4: 279-295

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

Veron M., G. Ris, J. Musse (1976), Continuity of Biparametric Surface patches, ***Computer Aided Design*** 8(4):267-273

Watkins, M. A. (1988), Problems in Geometric Continuity, ***Computer Aided Design*** 20, No 8

Yamaguchi Fujio (1988), ***Curves and Surfaces in Computer Aided Geometric Design*** Springer Verlag, Berlin, Heidelberg, 1988