UNIVERSITY OF WISCONSIN-MADISON
CENTER FOR THE MATHEMATICAL SCIENCES

LOCAL INTERPOLATION OF A CUBIC CURVE MESH BY A PIECEWISE [BI]QUARTIC C^1 SURFACE WITHOUT SPLITTING

by
Jörg Peters

Technical Summary Report # 89-25
March 1989

→ Constructive Approximation

Abstract

We study the interpolation of a mesh of curves by a piecewise polynomial C^1 surface that consists of exactly one patch per mesh facet. In particular, we derive the precise necessary and sufficient condition on the mesh data to allow for interpolation. We apply the result by exhibiting an algorithm for the local interpolation of a cubic mesh by a piecewise [bi]quartic C^1 surface.

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

Key words: Bernstein-Bézier form, C^1 surface, interpolation, (i,j,k)-match, compatibility

This research was supported by NSF DMS-8701275

1. Introduction

Fitting a smooth parametric piecewise polynomial surface to a given mesh of curves is a well studied problem. A popular approach to its solution is based on 'blending' polynomial patches: each patch matches the data of one edge of a mesh hole and remains 'neutral' on the others so that the sum of the patches matches all the mesh data [Coons '67]. To obtain the desired smoothness, blending schemes need to know the normal derivative at each mesh point. This gives extra control but comes at a cost in the form of 'compatibility constraints'. There is one such constraint on the mesh data at each patch corner. A different approach was championed by Sarraga [Sarraga '86]. His scheme does away with the normal derivatives and aims at just interpolating the mesh and no further information. That is, he does not assume that the 'continuity links' between adjacent patches have been removed a priori by prescribing the common normal derivatives. The additional complexity of this approach is rewarded by fewer restrictions on the curve mesh: at points with an odd number of neighbors, the curves can be arbitrary; at points with four neighbors just one constraint suffices (Sarraga does not treat the general 'even-point' case). Our algorithm is in many aspects akin to Sarraga's. However, it includes the general even-point case, adds total degree (triangular) patches to the surface construction and employs, generically, *half* as many coefficients. This is possible by exploiting all the degrees of freedom not pinned down by the **compatibility constraints** and the **smoothness conditions**. In particular, we do not restrict our scheme to follow the well established sufficient but *not necessary* C^1 conditions laid out in [Farin '82 p.277] and [Farin '83 p.57]. Instead, we derive the necessary and sufficient smoothness conditions for polynomial patches from first principles. This includes a proof that, up to a common factor, the *three* 'weight functions' that relate the directional derivatives along and across a patch boundary are polynomials of low degree. We then show how to exploit the newly found flexibility by constructing a piecewise [bi]quartic (geometrically) C^1 surface that interpolates a given mesh of cubic boundary curves and associates exactly one polynomial patch with each facet. The construction is *local* and *linear*.

The primary point of this paper is the precise analysis of the conditions that guarantee that a polynomial curve mesh has a smooth interpolant, i.e. a complete characterization of the *compatibility constraints*. It turns out that these constraints are less stringent than they appear from the analyses of 'twist matrices' in [Sarraga '86] and [Watkins '88]. The additional freedom comes by including the scalar coefficients of the weight functions as variables. To cope with the one scalar constraint still imposed by the compatibility constraints, we present an effective way of locally generating compatible quartic mesh curves. This construction is based on a theorem that ascertains that curve meshes that are consistent with second order data at the mesh points always allow for interpolation. The same theorem explains why facet splitting at the centroid resolves the compatibility problem, while rational patches only help if the polynomial in the denominator vanishes at all the interpolation points. Finally, we establish that the number of sides of the facets at a mesh point has no bearing on compatibility: the additional difficulties in interpolating cubic meshes by triangular quartic patches are solely due to the low number of coefficients of those patches.

To avoid ambiguities, we will adhere to the following notation. A match between an i-sided and a j-sided patch is called an **i-j configuration**. For example, if two biquartic, hence 4-sided, patches share an edge (not just a point), we refer to this as a 4-4 configuration. Since we use exactly one piece per facet, we only encounter 3-3, 3-4 and 4-4 configurations. The intersections of the mesh curves will be called **data points** or just **points**. We define a **j-point** to be a data point with j incident mesh curves. Correspondingly, an **even-point** is a data point with an even number of incident mesh curves. For easy reference and so that the reader will not be disappointed later, we list the assumptions of our construction already at the outset. First, the algorithm expects consistency of the data: the tangent planes at the points must be well defined and the mesh facets triangular or rectangular. Secondly, since 3-3 configurations offer fewer degrees of freedom and we want to remain within the quartic setup, we assume symmetry in the data. This allows us to apply a 'simple' C^1 match. If one or both of the mesh cells may be split, the assumption can be dropped. Finally, we subject the data at even-points to a compatibility constraint *sufficient* to guarantee that a [bi]quartic or higher degree C^1 surface can be fitted into the mesh without splitting. In its first variant, i.e. as Assumption 1.3.a, the constraint applies only to 4-points and is due to [Sarraga '86]. The second variant, Assumption 1.3.b, applies to general even-points. The complete necessary and sufficient condition, phrased in terms of the curvature components of the mesh curves, are derived in Section 3. We now give the list in formal terms.

Assumption 1.1. *[basic consistency]*
(a) The mesh curves define unique tangent planes at the data points.
(b) The mesh has only three- and four-sided facets.
(c) The data are well-distributed, i.e. at each mesh point (with a given or approximated normal), the projections into the tangent plane of any two edges belonging to the same facet span an angle of less than π.

Assumption 1.2. *[3-3 symmetry] For each data point lying on the common mesh curve of a 3-3 configuration, express the first difference vector of the BB form as a weighted average of the difference vectors of the left and the right neighbor curve and form the ratio of the two weights. The ratios at the two points on the common mesh curve are equal.*

Assumption 1.3. *[even-point compatibility] At every even-point P one of the following holds.*
(a) For each mesh curve incident at P, the tangent vectors of the two neighboring curves are collinear.
(b) There exists a second fundamental form at P to which all mesh curves conform.

To set the algorithm into perspective, we introduce a classification of local surface constructions that are based on polynomial patches. As Section 2 will show, the C^1 matching constraints for patches consist of *non-local nonlinear equalities and inequalities*. Hence an important feature of any surface construction is its selection of (geometrically meaningful) variables that can be fixed a priori, that is input or derived from the data, so as to arrive at a sufficient and consistent set of preferably linear constraints in the remaining variables. Consequently, we sorted the schemes in Table 1 below according to the interpolated data.

The first set of algorithms, to which our algorithm belongs, interpolates a mesh of curves. The second group consists of blending methods that match normal derivatives in addition to the mesh. The third group prescribes the normal direction along patch boundaries, and the fourth group is set apart since its algorithms create additional free variables by splitting the original mesh facets. We caution that the table below only reflects the author's current knowledge of the field, hence may be incomplete.

Table 1: Local Piecewise Polynomial Surface Interpolation Schemes

Interpolated data	Facet shape	Patches per facet	Degree of patches	Continuity	References
P N T M	4	1	6	1	Sarraga '86
P N T M	3,4	1	4	1	Peters '89
P N T M D	4	1	3	1^0	Coons '67
P N T M D	3	1	$(9,9)^r$	1	Barnhill ea '73
P N T M D	3,4	1	$(7,4)^r$	1	Gregory '74
P N κ	4	1	3	1^0	Sabin '68
P N (κ)	3,4	1	3	1^0	Peters '88a
P N	3	3	4	1	Farin '83
P N	n	$\geq 2n^1$	5 ($\frac{4}{1}$) 11	1^0 (2^0)	Jones '88
P N T	3	3	4	1	Piper '87
P N T M	3,4	3,4	4	1	Shirman ea '87
P N T M	3	3	3	1	Peters '88b

* P = location, N = normal, T = tangent, κ =curvature, M = cubic curve mesh
D = normal derivative
[0] the scheme has restrictions beyond those of Theorem 3.2 of this paper
[1] ~~neighboring facets have to be split~~ only edge-pts
[r] the blending functions are rational (numerator, common denominator)

Our algorithm is thus characterized by observing that the resulting surface interpolates a cubic curve mesh, associates a 3- or 4-sided facet with one polynomial patch, and constructs, for appropriate data, a piecewise maximally [bi]quartic C^1 surface. The challenge faced by Sarraga's and this algorithm is to get by without splitting facets. This means fewer patches, but increases the degree and imposes restrictions on the curve mesh. The splitting scheme described in [Peters '88b], for comparison, can interpolate (almost) arbitrary meshes using three *cubic* patches. In fact, given the restrictions that have to be applied to the cubic curve mesh, and the ability of 'splitting' approaches to remove them, we suggest and show how the basic algorithm can be made more flexible by allowing it to split facets (cf. Sections 3.3 and 4.3).

We represent the polynomial surface patches in Bernstein-Bézier form (**BB form**). Besides stability under evaluation and differentiation, this form gives geometric meaning to its coefficients and easy access to value and derivative information along patch boundaries (cf. [Farin '86] and [de Boor '87]). To construct the surface, we determine the coefficients as 3-vectors, i.e. the polynomial pieces map from the unit square or triangle to 3-space. In our derivation and analysis, it will suffice to look at univariate polynomials, i.e. derivatives of the patches evaluated at a boundary. Since our main algebraic work consists of multiplying these univariate polynomials, we find it advantageous to use an abbreviation which

explicitly lists the scalar coefficients $\binom{d}{j}$ together with the control points. That is, we use

$$\underbrace{(\alpha,\ldots,\beta,\ldots,\gamma)}_{d+1 \text{ terms}}$$

to denote the polynomial

$$t \mapsto \alpha(1-t)^d + \ldots + \beta(1-t)^{d-j}t^j + \ldots + \gamma t^d.$$

For example, raising the degree of the quadratic polynomial $(a^0, 2a^1, a^2)$ 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, this avoids writing fractions. Abbreviating the scalar product of a and b as ab keeps the notation simple, e.g. $(a^0, a^1)(n^0, n^1) = (a^0n^0, a^0n^1 + a^1n^0, a^1n^1)$. We use subscripts to count the curves emanating from a given point and superscripts to number the coefficients of a polynomial. The counting is cyclic, i.e. modulo the number of neighbors.

Sections and content:
In Section 2, we derive the conditions for a C^1 match between two surface patches in terms of the BB form, i.e. as an (i,j,k)-match. In particular, we show that if the patches are polynomial, then, without loss of generality, the degree of the 'weight functions' that relate the directional derivatives along and across a patch boundary can be bounded in terms of the patch degrees. In Section 3, we characterize the compatibility problem for surfaces and compare the results with those in [Sarraga '86] and [Watkins '88]. In Section 4, a technical section, we derive the pairs of configurations and matches appropriate for the quartic setup. These pairs are used in Section 5 to state the algorithm for coding. Section 6 presents an algorithm for constructing quartic curve meshes that are compatible and Section 7 concludes with some examples from our implementation.

2. The C^1 conditions

We now review the definition of a C^1 surface match and specialize it to patches in BB form. Consider a patch p parametrized by u and v and its neighbor patch, q, parametrized by u and w as depicted in Figure 2(a) below.

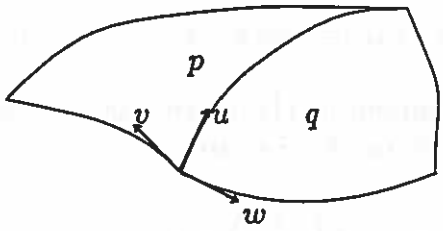


Figure 2(a): Parametrization of abutting patches.

We concentrate on the common boundary where $v = w = 0$. Assuming that both patches are sufficiently smooth we use the subscripts u, v and w to denote partial derivatives along the boundary, i.e. $p_u := \frac{\partial}{\partial u} p(u,0)$, $p_v := \frac{\partial}{\partial v} p(u,0)$ and $q_w := \frac{\partial}{\partial w} q(u,0)$. Then p and q are part of a C^1 surface if and only if the surface normal is well-defined on both patches and agrees at every point of the boundary:

$$\frac{p_v \times p_u}{\|p_v \times p_u\|} = \frac{p_u \times q_w}{\|p_u \times q_w\|}, \quad \text{[matching normal]} \tag{1}$$

$$p_v \times p_u \neq 0. \quad \text{[non-vanishing normal]} \tag{2}$$

Note that we were careful with the order in the vector products. The following equivalent characterization of the C^1 conditions for patches can be found, e.g. in [Peters '88b].

Lemma 2.1. [C^1 conditions] *A match between two smooth patches $p(u,v)$ and $q(u,w)$ across a common boundary parametrized by u is C^1 if and only if there exist scalar-valued functions λ, μ and ν of u such that, at each point of the boundary,*

$$\lambda p_u = \mu p_v + \nu q_w \quad \text{[common tangent plane]} \tag{E}$$

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

We call λ, μ and ν the **weight functions** of the match and use (E) to denote to the equality constraints and (I) when we refer to the inequality constraints.

Corollary 2.2 below characterizes the weight functions in case p and q are polynomials. It ascertains that, up to a common factor, the weight functions are also polynomials and that their degree is bounded in terms of the degree of the patches. A similar claim appears in [Liu '86]. To make our development precise, we define d^p to be the degree of the (univariate) polynomial p and set $d^u := d^{p_u}$, $d^v := d^{p_v}$ and $d^w := d^{q_w}$ to avoid double indices.

Corollary 2.2. [polynomial weight functions] *If p and q are polynomials, then, up to a common factor, λ, μ and ν in Lemma 2.1 are polynomials in u of degree no larger than $d^v + d^w$, $d^w + d^u$ and $d^u + d^v$ respectively.*

Proof. We denote the vector-valued polynomial corresponding to the i^{th} and j^{th} component of p_u by p_u^{ij} and define, for $i \neq k \neq j$ and $\{i,j,k\} = \{1,2,3\}$,

$$l_k := \det(p_w^{ij}, -p_v^{ij}) \quad m_k := \det(p_u^{ij}, p_w^{ij}) \quad n_k := \det(p_v^{ij}, p_u^{ij}).$$

Since p_u and p_v are linearly independent by (I), there exists a k such that n_k is a nontrivial *polynomial.* Applying Cramer's rule to (E), i.e. to

$$(p_u, p_v)\begin{pmatrix} \lambda \\ -\mu \end{pmatrix} = \nu q_w,$$

we obtain the formal solution along the boundary as

$$\begin{pmatrix} \lambda \\ -\mu \\ \nu \end{pmatrix} = \frac{\nu}{n_k}\begin{pmatrix} l_k \\ m_k \\ n_k \end{pmatrix}. \tag{3}$$

It remains to show that all roots of n_k are also roots of l_k and m_k so that common factors can be cancelled and the solution is welldefined. For this it is sufficient to show that λ and μ are continuous at any root u of n_k. Since (I) implies that there exists an $l \neq k$ such that $n_l(u) \neq 0$ and hence a formula like (3) with k replaced by l holds at and in some neighborhood of u, the continuity is established. ♣

The C^1 conditions for abutting polynomial patches are often stated in the form

$$q_w = \alpha p_u + \beta p_v.$$

From this formulation one can easily arrive at the conclusion that mathematics or simplicity demand that α and β have to be polynomials. However, as the lowest degree schemes listed in Table 1 demonstrate, this conclusion is wrong. Rather, Lemma 2.1 and Corollary 2.2 make clear that α and β are in general (even after removing a common factor) rational functions, i.e. a polynomial divided by a polynomial. The assumption that α and β are polynomials is equivalent to assuming that either μ or ν is constant; hence we will refer to it as the 'constancy assumption'. To illustrate the prevalence of the constancy assumption, we list some papers that derive C^1 conditions or apply such conditions in the construction of surfaces. Most references initially state the C^1 conditions in full generality, i.e. as continuity of the surface normal or as continuity of the appropriate connecting diffeomorphism. However, once the weight functions are derived in the BB setup, one of the weight functions is treated as a constant. We looked at [de Rose '85 p.59], [Farin '82 p. 277], [Farin '83 p. 50,57], [Farin '88 p. 248,250], [Faux, Pratt '79 p. 216], [Hahn '87 p. 14], [Höllig '86 p.14], [Jones '88 p. 329], [Sabin '77 p.85], [Sarraga '86 p.5], [Shirman, Sequin '87 p.285] [van Wijk '84 p.4]. [Liu '86 p. 438] also proposes constant coefficient matches for

the surface construction, but additionally gives an example of a surface continuation based on 'non-constant' smoothness conditions. More generally, the cubic mesh interpolation in [Peters '88b] shows that it is not only correct but also very useful to explore the possibility of having *all three weight functions non-constant.* The C^1 matches are correspondingly characterized by the triple (**degree of λ, degree of μ, degree of ν**). With this notation, we can interpret Piper's solution [Piper '86 p.227] as a (2,1,1)-match and the construction of [Chiyokura, Kimura '83 p.295] as a (1,1,1)-match. An explicit statement of the cusping conditions in terms of the BB form is also useful. Piper, in his reference, poses the question as to whether his construction is free of cusps. A close look at the inequality constraints of Lemma 2.1 proves that Assumption 1.1.c is both necessary and sufficient for his and the construction in [Peters '88b] to be free of cusps. The appropriate 'non-constant' cusp-free (i,j,k) C^1 matches for our surface construction are derived in Section 4.

3. The 'corner compatibility problem'

For 'blending' approaches [Coons '67], it is well known (see e.g. [Gregory '74]) that Assumption 1.a, i.e. the existence of a well defined tangent plane at the data points, is not sufficient to guarantee the existence of a C^1 mesh interpolant. In particular, since the blending approaches prescribe 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 constraints of the type $p_{uv} = p_{vu}$ at P.

However, as Sarraga points out, for rectangular patches in BB form [Sarraga '86] and Watkins, for rectangular patches of arbitrary polynomial representation [Watkins '88], curve meshes by themselves need only satisfy one constraint per even-point and none at odd-points. In particular, both authors draw attention to a certain matrix that arises in determining 'twist coefficients', the major step in the interpolation process. This matrix is always invertible at odd-points, but rank deficient at even-points. Related observations on the influence of parity on the solvability of a crucial set of equations appear in in [van Wijk '83] and [Peters 88a]; hence the interest in fully understanding the problem. For 4-points, Sarraga exhibits an easy-to-check sufficient condition listed here as Assumption 1.3.a. The challenge is to constrain the right hand side of the rank deficient system just right to make it solvable. Unfortunately, this condition and thus Sarraga's construction for even-points applies solely to points with exactly four neighbors. For arbitrary even-points and non-rectangular patches, the compatibility question: "how must a curve mesh be constrained to allow for interpolation by a surface" was left unanswered.

This is where Lemma 3.1 below 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 heart of the analysis, however, is Theorem 3.2. Theorem 3.2 gives the precise necessary and sufficient condition that guarantees that a quartic or higher degree polynomial patch can be smoothly fitted into a mesh of cubic curves. The condition is weaker than Sarraga's (one 'scalar constraint' *vs* one 'vector constraint') and indicates that

(a) the compatibility constraints at any point are independent of just how many edges there are to the impinging patches, and that

(b) the compatibility constraints apply to rational (polynomial) patches in the same fashion as to polynomial patches unless the denominator vanishes at all data points.

3.1 Derivation of the 'twist equations'

By Corollary 2.2, the weight functions are, without loss of generality, polynomials, and hence differentiable. The gist of the compatibility problem can be captured by differentiating the C^1 equality constraints,

$$\lambda p_u = \mu p_v + \nu q_w, \qquad (E)$$

with respect to u and evaluating the result at $u = 0$, i.e. at a data point P:

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

The data are 'compatible' if the system of constraints that arises from collecting (E_u) for each mesh curve incident to P, is solvable.

For mnemonic efficiency, we denote the BB coefficients of p_u, p_v and q_w by u^i, v^i and w^i. The terms that appear in (E_u) are then given by

$$(\lambda^0, d^\lambda\lambda^1, \ldots)(u^0, d^u u^1, \ldots) = (\mu^0, d^\mu\mu^1, \ldots)(v^0, d^v v^1, \ldots) + (\nu^0, d^\nu\nu^1, \ldots)(w^0, d^w w^1, \ldots), \qquad (E')$$

where $d^\lambda + d^u = d^\mu + d^v = d^\nu + d^w$. (We do not need to worry about the meaning of the λ^1, u^1, μ^1, etc. terms in case these polynomials are just constants, since then $d^\lambda = d^u = \ldots = 0$.) To obtain the matching constraints at P, we simply set the coefficients of the scalar product polynomials with respect to the BB basis to zero. This yields the **tangent constraints**

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

and the **twist constraints**

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

We now look at all n mesh curves incident to P numbering the patches and curves clockwise. With $\alpha > 0$ (to comply with (I)) and scalar, the tangent constraints are equivalent to

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

Having thus determined all η_i and k_i, the twist equations become

$$d_i^u\alpha_i\eta_i u_i^1 + d_i^\lambda\lambda_i^1 u_i^0 = d_i^v\alpha_i k_i v_i^1 + d_i^\mu\mu_i^1 v_i^0 + d_i^w\alpha_i(1-k_i)w_i^1 + d_i^\nu\nu_i^1 w_i^0. \qquad (E_1')$$

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. 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, up to the common factor α, pinned down by solving the tangent equations at the neighbor point P_i. Similarly, if the degree of p_v or q_w is low, v^1 and w^1 could be determined at this stage. Cubics, for example, have overlapping twist constraints, while bicubics have separated ones. Next, we raise the degree of any two adjacent boundary curves to the same value, i.e., with $p_i(v) := p_u(0, v)$ and $p_{i+1}(u) := p_v(u, 0)$, to $d_i^m := \max(d^{p_i}, d^{p_{i+1}})$. This avoids the purely notational problems that arise, e.g. in the analysis of cubic-by-sextic patch complexes. With the cubic-by-sextic patches raised to a bisextics, we have achieved that the interior Bézier coefficient closest to P in the i^{th} patch is welldefined. We call this coefficient the 'twist coefficient' (for historical reasons) and denote it by t_i. A comparison of the degrees of freedom with the number of equations leads to the following approach: pin down the scalar variables by some rule and try to solve for the twist coefficients. That is, solve

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

where

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

and $B_i := u_i^0 + P$, the Bézier coefficient on the boundary curve closest to P.

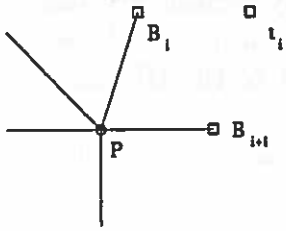


Figure 3(a): Orientation at a data point

The twist constraints at P then consist of 3 independent n by n systems – one for each coordinate. Each system is of the form

$$KT = R, \qquad (T)$$

where

$$K := \begin{pmatrix} d_1^m k_1 & d_2^m(1-k_1) & \dots & 0 & 0 \\ 0 & d_2^m k_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \dots & d_{n-1}^m k_{n-1} & d_n^m(1-k_{n-1}) \\ d_1^m(1-k_n) & 0 & \dots & 0 & d_n^m k_n \end{pmatrix},$$

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

Unfortunately, as Lemma 3.1 shows, (T) is not always solvable given that the contribution of the u^1 terms in R is arbitrary.

Lemma 3.1. [invertibility at odd-points] *The matrix in (T) is of full rank if and only if P is an odd-point. Otherwise its rank is deficient by one.*

Proof. Looking at the u_i^0 as 2-vectors in the tangent plane, we have from (E_0') for the i^{th} edge:

$$k_i = \frac{\det(u_i^0, u_{i+1}^0)}{\det(u_i^0, u_{i+1}^0 - u_{i-1}^0)}.$$

On the other hand, for any n-vector $t \in \ker(K)$

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

and thus

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

Since

$$\prod_{i=1}^{n} \frac{k_i d_i^m}{(k_i - 1) d_{i+1}^m} = \prod_{i=1}^{n} \frac{k_i}{k_i - 1} = \prod_{i=1}^{n} \frac{\frac{\det(u_i^0, u_{i-1}^0)}{\det(u_i^0, u_{i+1}^0 - u_{i-1}^0)}}{\frac{\det(u_i^0, u_{i+1}^0)}{\det(u_i^0, u_{i+1}^0 - u_{i-1}^0)}} = (-)^n,$$

the kernel is nontrivial if and only if n is even. ♣

3.2 The 'extended twist equations'

Lemma 3.1 makes clear that our only hope for solving (E_u) at even-points lies in adjusting 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 is a reduction from one vector-valued constraint (Lemma 3.1) to one scalar-valued constraint (Theorem 3.2). This constraint has an easy-to-check sufficient condition in Assumption 1.3.

Theorem 3.2. [sufficiency of C^2 data] *In general, system (T') is inconsistent for even-points. However, Assumption 1.3 is sufficient to guarantee a solution.*

Proof. We first restrict our attention to the λ_i^1 terms and show later that the analysis remains the same if we allow additional scalar coefficients to vary. With u_i^x be the first component of u_i^0, $R'^x(i)$ 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 and

$$U^x := \begin{pmatrix} -\frac{d_1^\lambda}{\alpha_1} u_1^x & 0 & \dots & 0 \\ 0 & -\frac{d_2^\lambda}{\alpha_2} u_2^x & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -\frac{d_n^\lambda}{\alpha_n} u_n^x \end{pmatrix} \tag{4}$$

(E_u) becomes

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

We show that (T′) is not always solvable. To simplify the analysis, we transform the coordinate system rigidly, preserving orientation and angle, so that P is mapped to the origin and the normal N at P to (0, 0, 1). This preserves the block structure of (T′):

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

$$\begin{pmatrix} k_1 & (1-k_1) & 0 & \lambda\bar{v}_1 \\ & k_2 & (1-k_2) & \lambda\bar{v}_2 \\ & & k_3 & \lambda_3\bar{v}_3 \\ (1-k_4) & & & \lambda_4\bar{v}_4 \end{pmatrix} \qquad k_1k_2k_3\lambda_4\bar{v}_4 \quad -(1-k_4)$$

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. Consequently, we focus on $K_3T_3 = R_3$. The only contributions to R_3 comes from the $\lambda^0 u_{i3}^1$ terms (cf. (E_1'')), where the second subscript '3' indicates the non-tangential component. All other terms are vectors in the tangent plane and hence appear only in R_1 and R_2. Since K_3 is of rank $n-1$, (T'') is solvable exactly if R_3 is in the span of the first $n-1$ columns of K_3. That is, if and only if,

$$\det \begin{pmatrix} d_1^m k_1 & d_2^m(1-k_1) & \dots & 0 & \lambda_1^0 u_{13}^1 \\ 0 & d_2^m k_2 & \dots & 0 & \lambda_2^0 u_{23}^1 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \dots & d_{n-1}^m k_{n-1} & \lambda_{n-1}^0 u_{n-1,3}^1 \\ d_1^m(1-k_n) & 0 & \dots & 0 & \lambda_n^0 u_{n3}^1 \end{pmatrix} = 0. \qquad (5)$$

Assumption 1.3.a implies (5) by stipulating that λ_i^0 be zero. To show the sufficiency of Assumption 1.3.b we look directly at (E_u). Since there exists a symmetric matrix Q of rank two such that $Ns_{uu} = -s_uQs_u$, the normal component of (E_u) at $u = 0$ is

$$-\lambda s_uQs_u = \mu Ns_{uv} + \nu Nt_{uw}. \qquad (E_u^N)$$

If we choose the normal component of each t_i so that s_{vu} satisfies $Ns_{uv} = -s_uQs_v$ at the data point, equation (E_u^N) reduces to

$$(\lambda s_u - \mu s_v - \nu t_w)Qs_u = 0 \qquad \text{at} \quad u = 0$$

which is already implied by the tangent constraints (E_0).

We could try to make use of the other scalar variables in (E_1). However, since the corresponding vectors u_i^0, v_i^0 and w_i^0 all lie in the same (tangent) plane, we cannot improve the result. ♣

3.3 Implications of Theorem 3.2

A look at the proofs of Lemma 3.1 and Theorem 3.2 confirms that compatibility at a point is independent of just how many edges there are to the impinging patches. We record this fact in the following corollary.

Corollary 3.2.a. *The compatibility problem exists for any combination of 3- or 4-sided or even n-sided patches.*

Next we consider the problem of immediate interest to the practitioner: how can one obtain 'compatible' meshes? We discuss some alternatives. If $d_1^m = \dots = d_n^m$, then (5) can be expressed as

$$\lambda_n^0 u_{n3}^1 = \sum_{j=1}^{n-1}(-)^{j+1}\Big(\prod_{l=1}^{j}\frac{1-k_{l-1}}{k_l}\Big)\lambda_j^0 u_{j3}^1. \qquad (6)$$

Since 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, this suggests adjusting the u_{n3}^1. For cubic boundaries, however, any change in u_{n3}^1 has also an impact on the equations at P_n. A better approach seems therefore the local and symmetric enforcement of Assumption 1.3.b. However, local interpolation of positional, normal and curvature data by a cubic, as discussed in [Sabin '68], [de Boor, Höllig, Sabin '87] and [Peters 88c], is not always feasible. This leaves us with the alternative to use global cubic spline interpolation for 'regular' meshes or to shift to quartic mesh curves. The latter idea is pursued in Section 5.1.

Next, we establish that Assumption 1.3 is in general not necessary. For this recall that the second fundamental form consists of three pieces of information. 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 the form; the fourth curve then has to satisfy a single scalar constraint. But, even though this constraint implies (6) and acts on the same variables as (6), it is not equivalent.

Claim 3.2.b. *The existence of a solution to the compatibility problem at a 4-point does not imply Assumption 1.3.*

We prove the claim by exhibiting data that can be extended to a set of compatible data, but neither satisfies the collinearity nor the curvature condition of Assumption 1.3.
Example 3.3. We choose $N = (0,0,1)$ $u_1^0 = (1,0,0)$, $u_2^0 = (0,-1,0)$, $u_3^0 = (-1,1,0)$, $u_4^0 = (1,1,0)$, $\alpha = 1$. Then

$$k_1 = \frac{1}{2}, \lambda_1^0 = \frac{1}{2}, k_2 = \frac{1}{2}, \lambda_2^0 = -\frac{1}{2}, k_3 = \frac{1}{3}, \lambda_3^0 = -\frac{1}{3}, k_4 = \frac{2}{3}, \lambda_4^0 = \frac{1}{3},$$

and Assumption 1.3.a does not hold. Hence we focus on the curvature assumption, 1.3.b. On one hand, compatibility, (6), implies

$$r_4 = \beta_1 r_1 - (\frac{(1-k_4)(1-k_1)}{k_1 k_2})r_2 + \beta_3 r_3 = \beta_1 r_1 - \frac{2}{3}r_2 + \beta_3 r_3, \tag{7}$$

for $r_i := \lambda_i^0 u_{i3}^1$ and some constants β_1 and β_3. On the other hand, Assumption 1.3.b implies $-u_4^0 Q u_4^0 = N\bar{u}_4$. Hence, using the tangent equations, (E_0), we get

$$-((1-k_4)u_3^0 + k_4 u_1^0)Q((1-k_4)u_3^0 + k_4 u_1^0) = N\lambda_4^0 r_4$$

which are, by assumption, equal to

$$(1-k_4)^2 N\bar{u}_3 + k_4^2 N\bar{u}_1 - 2(1-k_4)k_4 u_1^0 Q u_3^0 = N\lambda_4^0 r_4.$$

We subtract the analogous equation for $i = 2$, i.e.
$(1-k_2)^2 N\bar{u}_1 + k_2^2 N\bar{u}_3 - 2(1-k_2)k_2 u_1^0 Q u_3^0 = N\lambda_2^0 r_2$ and eliminate Q to obtain

$$\begin{aligned}(1-k_4)k_4\big((1-k_2)^2 N\bar{u}_1 + k_2^2 N\bar{u}_3\big) - (1-k_2)k_2\big((1-k_4)^2 N\bar{u}_3 + k_4^2 N\bar{u}_1\big) \\ = N(1-k_4)k_4\lambda_2^0 r_2 - N(1-k_2)k_2\lambda_4^0 r_4.\end{aligned} \tag{8}$$

It remains to show that (7) does not imply (8), hence does not imply Assumption 1.3.b. We replace r_4 in (8) and collect the terms associated with Nr_2. Since r_2 is unrestricted (we can always use r_3 or r_1 to enforce (6)), the coefficient of Nr_2 must vanish. However,

$$(1-k_4)k_4\lambda_2^0 - \frac{2}{3}(1-k_2)k_2\lambda_4^0 = -\frac{1}{18} \neq 0.$$

♣

Finally, we consider *rational polynomial patches*, i.e. patches of the form S/s where S is a vector-valued and s a scalar-valued polynomial. We show

Corollary 3.2.c. *The compatibility problem exists for rational polynomial patches unless the polynomial in the denominator vanishes at all data points.*

Proof. Since Gregory's patches (cf. [Gregory'74], [Chiyokura '83]) are in rational form where S is biseptic and s biquartic for 4-sided patches and the degree is 7 and 3 for 3-sided patches, and since these patches match arbitrary cross derivatives at the data points, compatibility is not a problem. Conversely, to obtain the advantage of the rational setup, s must vanish at the data points. For, if $s(0) \neq 0$, we may assume, after scaling and shifting, that $s(0) = 1$ and $S(0) = 0$. Differentiating the rational equivalent of (E),

$$\lambda\frac{S_u s + s_u S}{s^2} = \mu\frac{S_v s + s_v S}{s^2} + \nu\frac{T_w t + t_w T}{t^2}, \qquad (E^{rat})$$

with respect to u, we obtain

$$\lambda_u\frac{S_u s + s_u S}{s^2} + \lambda\Big(\frac{S_{uu}s + 2s_u S_u + s_{uu}S}{s^2} - 2s_u\frac{S_u s + s_u S}{s^3}\Big) = \lambda_u S_u + \lambda S_{uu} =$$

$$\mu_u S_v + \mu(S_{vu} + s_v S_u - s_u S_v) + \nu_u T_w + \nu(T_{wu} + t_w S_u - s_u T_w).$$

That is, we arrive at the same setup that we discussed in Lemma 3.1 and Theorem 3.2.

♣

A look at the C^1 equality constraints for rational polynomial patches, (E^{rat}), shows that the high degree of the Gregory patches is in the nature of the construction: the degree of the numerator has increased to $d^S + d^s - 1$. For example, if the weight functions μ and ν are constant, then the number of vector equations rises to $d^S + d^s$ *vs* d^S in the polynomial case; yet there are only d^s additional *scalar* degrees of freedom.

3.4 Applications to splitting and Sarraga's scheme

We will now examine why *splitting*, i.e. the partitioning of one triangle into three subtriangles at the centroid, resolves the compatibility problem even though it creates even-points. A partial explanation is provided by observing that the splitting technique generates the coefficients of the new interior boundaries in dependence on the twist coefficients and not vice versa (cf. [Farin 83], [Piper '87], [Peters '88b], [Shirman, Séquin 87]). The other crucial ingredient is the fact that the new coefficients are chosen to be straight

averages of their surrounding coefficients. Assuming that the twist coefficients, t_i, have been chosen in accordance with (E_1) at the original boundaries and that the degree of all triangles is the same, say d, we only have to check that (E_1'') holds at the new boundaries:

$$dk_i t_{i-1} + d(1-k_i)t_i = r_i. \qquad (E_1^S)$$

The averaging construction implies that $\lambda = 3/2$, $\mu = \nu = 1/2$ and $\alpha = 1$ for any new boundary so that $d^\lambda = d^\mu = d^\nu = 0$. This simplifies r_i (cf. (E_1'')) to

$$r_i = 1.5du_i^1 + \mu_i^0 dB_i + \nu_i^0 dB_i.$$

Dividing by $k_i = 1 - k_i = \mu_i^0 = \nu_i^0$ and d, we obtain

$$t_{i-1} + t_i = 3u_i^1 + 2B_i. \qquad (E_1^S)$$

By construction, $u^1 = (1/3)\big((t_{i-1} - B_i) + (t_i - B_i)\big)$, so that compatibility is achieved.

[Sarraga '86] offers a different approach to the compatibility problem than we are about to give. It is based on the fine observation that, if we choose the degree of the patches sufficiently high, then we can set the coefficients of the troublesome terms in (E_u) to zero (see also [Hahn '87] for a similar approach). This leads to particularly simple twist equations, as we will now show. We focus on 4-points which, by Sarraga's assumption, have $\lambda_i^0 = 0$. The nontangential u^1 terms (cf.(E_1'')) are removed by setting also the second term of λ to zero, so that $\lambda = \{0, 0, \lambda^2, \ldots\}$. If $\mu^0 \neq \mu^1$, the algorithm selects μ so that $\frac{\partial}{\partial u}\mu_u(0) = \frac{\partial}{\partial u}\mu_u(1) = 0$, i.e. $\mu := \{\mu^0, 3\mu^0, 3\mu^1, \mu^1\}$. This eliminates the v^0 terms on the right hand side. However, since s_v and t_w are at least cubic, and the algorithm is based on the constancy assumption, $\nu = 1$, this raises the degree of t_w to six: $d^w + d^\nu = d^w = d^v + d^\mu \geq 3 + 3$. Only for $\mu^0 = \mu^1$ can $d^w = 3$ be achieved. Altogether (T) now simplifies to the rank-deficient *homogenous* systems

$$0 = 3\mu_i^0 v_i^1 + 6w_i^1, \qquad i \in \{1, \ldots, 4\} \qquad (T^{S1})$$

for $\mu^0 \neq \mu^1$, ($\lambda^0 = \lambda^1 = \mu^1 = d^\nu = 0, \nu^0 = 1, d^w = 6, d^v = 3$), and

$$0 = 6\mu_i^0 v_i^1 + 6w_i^1, \qquad i \in \{1, \ldots, 4\} \qquad (T^{S2})$$

for $\mu^0 = \mu^1$, ($\lambda^0 = \lambda^1 = d^\mu = d^\nu = 0, \nu^0 = 1, d^w = 6, d^v = 6$).

4. Configurations and matches

In this 'technical', yet important section, we concentrate on the C^1 equality conditions for a cubic boundary mesh:

$$\lambda\widetilde{u} = \mu\widetilde{v} + \nu\widetilde{w} \quad \text{where} \quad \widetilde{u} = \{u^0, 2\bar{u}, u^1\}.$$

Choosing patches of degree 4, we keep the compatibility systems separate (cf. Section 3). For each configuration we derive two types of (i,j,k)-matches depending on the ratios k_0 and k_1 that we obtain from the tangent constraints:

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

If $k_0 \neq k_1$, we need a non-constant weight function for at least one of s_v or t_w, i.e. $d^\mu + d^\nu > 0$. We will restrict ourselves to $\alpha = 1$, except for 3-4 configurations. The analysis of the 3-3 configuration with $k_0 \neq k_1$ suggests how to fit an irregular n-gon with quartic triangular patches, a problem raised in [Shirman, Séquin 87].

4.1 The 4-4 configuration

The case $k_0 \neq k_1$ is the paradigm of our construction. With

$$\widetilde{v} =: \{v^0, 4v^{01}, 6\bar{v}, 4v^{10}, v^1\}$$
$$\widetilde{w} =: \{w^0, 4w^{01}, 6\bar{w}, 4w^{10}, w^1\}$$

we have the (3,1,1)-match $\lambda =: \{\lambda^0, 3\lambda^{01}, 3\lambda^{10}, \lambda^1\}$, $\mu =: \{\mu^0, \mu^1\}$, $\nu =: \{\nu^0, \nu^1\}$. Setting each coefficient with respect to the BB basis to zero, we obtain the following conditions for a C^1 match.

$$\begin{aligned}
\lambda^0 u^0 &= \mu^0 v^0 + \nu^0 w^0 && (E_0)\\
3\lambda^{01} u^0 + 2\lambda^0 \bar{u} &= 4\mu^0 v^{01} + \mu^1 v^0 + 4\nu^0 w^{01} + \nu^1 w^0 && (E_1^{44})\\
3\lambda^{10} u^0 + 6\lambda^{01} \bar{u} + \lambda^0 u^1 &= 4\mu^1 v^{01} + 6\mu^0 \bar{v} + 4\nu^1 w^{01} + 6\nu^0 \bar{w} && (E_2^{44})\\
3\lambda^{01} u^1 + 6\lambda^{10} \bar{u} + \lambda^1 u^0 &= 4\mu^0 v^{10} + 6\mu^1 \bar{v} + 4\nu^0 w^{10} + 6\nu^1 \bar{w}\\
3\lambda^{10} u^1 + 2\lambda^1 \bar{u} &= 4\mu^1 v^{10} + \mu^0 v^1 + 4\nu^1 w^{10} + \nu^0 w^1\\
\lambda^1 u^1 &= \mu^1 v^1 + \nu^1 w^1
\end{aligned}$$

The tangent equations (E_0) and the compatibility equations (E_1) determine all the coefficients except for $\bar{v}$ and $\bar{w}$. These we obtain by enforcing the conditions of type (E_2^{44}):

$$\begin{aligned}
\mu^0 \bar{v} + (1-\mu^0)\bar{w} &= \frac{1}{6}(3\lambda^{10} u^0 + 6\lambda^{01} \bar{u} + \lambda^0 u^1 - 4\mu^1 v^{01} - 4\nu^1 w^{01})\\
\mu^1 \bar{v} + (1-\mu^1)\bar{w} &= \frac{1}{6}(3\lambda^{01} u^1 + 6\lambda^{10} \bar{u} + \lambda^1 u^0 - 4\mu^0 v^{10} - 4\nu^0 w^{10}).
\end{aligned} \tag{9}$$

$(2\bar{\lambda}+\lambda^0)\,v^0 + 6$
$(2\bar{\lambda}+\lambda^1)\,v^1 +$

degree raised:
$(\lambda^0, \lambda^0+2\bar{\lambda}, \lambda^1+2\bar{\lambda}, \lambda^1)$ (μ,μ) (ν,ν)

Since $\mu^0 \neq \mu^1$, by assumption, we can solve for $\bar{v}$ and $\bar{w}$.

If, however, $k_0 = k_1$, we choose a (2,0,0)-match with $\lambda =: \{\lambda^0, 2\bar{\lambda}, \lambda^1\}$ and $\mu =: \{\mu\}$, $\nu =: \{\nu\}$. This leads to the system

$$\begin{aligned}
\lambda^0 u^0 &= \mu v^0 + \nu w^0 && (E_0)\\
2\bar{\lambda} u^0 + 2\lambda^0 \bar{u} &= 4\mu v^{01} + 4\nu w^{01} && (E_1'^{44})\\
\lambda^1 u^0 + 4\bar{\lambda}\bar{u} + \lambda^0 u^1 &= \mu \bar{v} + \nu \bar{w} && (E_2'^{44})\\
2\bar{\lambda} u^0 + 2\lambda^0 \bar{u} &= 4\mu v^{01} + 4\nu w^{01}\\
\lambda^1 u^1 &= \mu v^1 + \nu w^1
\end{aligned}$$

Again, $\bar{w}$ and $\bar{v}$ are undetermined after enforcing (E_0) and (E_1). However, this time there is only one equation. We enforce this equation by solving

$$\begin{aligned}
&\min \quad \frac{1}{2}(\|v^* - \bar{v}\|^2 + \|w^* - \bar{w}\|^2)\\
&\text{s.t. } \lambda^1 u^0 + 4\bar{\lambda}\bar{u} + \lambda^0 u^1 = \mu \bar{v} + \nu \bar{w}. \qquad (10)
\end{aligned}$$

As usual, we use degree raising or averaging to obtain the estimates $v^* := 4v^{01} + 4v^{10} - (v^0 + v^1)$ and $w^* := 4w^{01} + 4w^{10} - (w^0 + w^1)$.

Remarks:
(a) The scalar variable $\bar{\lambda}$ is shared by (E_1^{44}) at both endpoints of an edge. In general, this does not invalidate the approach taken in the even point case, since the number of incident edges is at least four. That is, we can associate two $\bar{\lambda}$'s exclusively with each data point, say P, and solve in such an order that all the $\bar{\lambda}$'s that are not associated with P are determined before solving for P.
(b)If $k_0 = k_1$ and $\bar{\lambda} = \lambda^0 + \lambda^1$, then (10) enforces a 'degree raised' cubic solution.

4.2 The 3-4 configuration

Let $\tilde{v}$ correspond to a triangular patch:

$$\begin{aligned}
\tilde{v} &=: \{v^0, 3v^{01}, 3v^{10}, v^1\}\\
\tilde{w} &=: \{w^0, 4w^{01}, 6\bar{w}, 4w^{10}, w^1\}.
\end{aligned}$$

We choose the (2,1,0)-match $\lambda =: \{\lambda^0, 2\bar{\lambda}, \lambda^1\}$, $\mu =: \{\mu^0, \mu^1\}$, $\nu =: \{\nu\}$. This leads to the system of equations

$$\begin{aligned}
\lambda^0 u^0 &= \mu^0 v^0 + \nu w^0 && (E_0)\\
2\bar{\lambda} u^0 + 2\lambda^0 \bar{u} &= 3\mu^0 v^{01} + \mu^1 v^0 + 4\nu w^{01} && (E_1^{34})\\
\lambda^1 u^0 + 4\bar{\lambda}\bar{u} + \lambda^0 u^1 &= 3\mu^1 v^{01} + 3\mu^0 v^{10} + 6\nu \bar{w} && (E_2^{34})\\
2\bar{\lambda} u^1 + 2\lambda^1 \bar{u} &= 3\mu^1 v^{10} + \mu^0 v^1 + 4\nu w^{10}\\
\lambda^1 u^1 &= \mu^1 v^1 + \nu w^1
\end{aligned}$$

After solving (E_0) and (E_1), equation (E_2^{34}) is enforced by setting

$$6\nu\bar{w} = \lambda^1 u^0 + 4\bar{\lambda}\bar{u} + \lambda^0 u^1 - 3\mu^1 v^{01} - 3\mu^0 v^{10}. \tag{11}$$

Remarks:
(a) The scalar variable $\bar{\lambda}$ is shared at both endpoints of an edge. (Cf. Remark (a) in Section 4.1.)
(b) The weights computed by (E_0') have to be properly scaled, so that ν agrees at both end points.

4.3 The 3-3 configuration

This section explains the role of Assumption 1.2, namely why 3-3 configurations have to be restricted in the quartic setup. First, we look at the case $k_0 = k_1$. Given that

$$\widetilde{v} =: \{v^0, 3v^{01}, 3v^{10}, v^1\}$$
$$\widetilde{w} =: \{w^0, 3w^{01}, 3w^{10}, w^1\}$$

the corresponding choices for the scalar weight polynomials are $\lambda =: \{\lambda^0, \lambda^1\}$, $\mu =: \{\mu^0\}$, and $\nu =: \{\nu^0\}$. That is, we choose a (1,0,0)-match. Setting each coefficient to zero, we have the following conditions for a C^1 match:

$$\begin{aligned} \lambda^0 u^0 &= \mu^0 v^0 + \nu^0 w^0 && (E_0) \\ \lambda^1 u^0 + 2\lambda^0 \bar{u} &= 3\mu^0 v^{01} + 3\nu^0 w^{01} && (E_1^{33}) \\ \lambda^0 u^1 + 2\lambda^1 \bar{u} &= 3\mu^0 v^{10} + 3\nu^0 w^{10} \\ \lambda^1 u^1 &= \mu^0 v^1 + \nu^0 w^1. \end{aligned}$$

Thus, for $k_0 = k_1$, the C^1 match is established once equations (E_0) and (E_1) are enforced.

Remark: There are no free λ's.

The case $k_0 \neq k_1$, however, has no easy solution.

Theorem 4.3. *A 3-3 configuration with quartics does not allow for a local C^1 match if $k_0 \neq k_1$.*

Proof. We show that the degrees of freedom in the quartic patches do not suffice for a (2,1,1)-match. Since no degrees of freedom are gained by going to higher order matches, the general result follows. Again we have

$$\widetilde{v} =: \{v^0, 3v^{01}, 3v^{10}, v^1\}$$
$$\widetilde{w} =: \{w^0, 3w^{01}, 3w^{10}, w^1\}.$$

The (2,1,1)-match implies $\lambda =: \{\lambda^0, 2\bar{\lambda}, \lambda^1\}$, $\mu =: \{\mu^0, \mu^1\}$ and $\nu =: \{\nu^0, \nu^1\}$. This leads to the system of equations

$$\begin{aligned}
\lambda^0 u^0 &= \mu^0 v^0 + \nu^0 w^0 && (E_0)\\
2\bar{\lambda} u^0 + 2\lambda^0 \bar{u} &= 3\mu^0 v^{01} + \mu^1 v^0 + 3\nu^0 w^{01} + \nu^1 w^0 && (E_1'^{33})\\
\lambda^1 u^0 + 4\bar{\lambda}\bar{u} + \lambda^0 u^1 &= 3\mu^1 v^0 + 3\mu^0 v^1 + 3\nu^1 w^0 + 3\nu^0 w^1 && (E_2'^{33})\\
2\bar{\lambda} u^1 + 2\lambda^1 \bar{u} &= 3\mu^1 v^{10} + \mu^0 v^1 + 3\nu^1 w^{10} + \nu^0 w^1\\
\lambda^1 u^1 &= \mu^1 v^1 + \nu^1 w^1.
\end{aligned}$$

Equation (E_0) leaves room only for positive scaling of the scalar coefficients. Since (E_0) and ($E_1'^{33}$) pin down $\widetilde{v}$ and $\widetilde{w}$ completely, there are, in general, not enough degrees of freedom to satisfy ($E_2'^{33}$). ♣

We propose to attack 3-3 configurations with $k_0 \neq k_1$ by centroidal splitting as in [Peters '88b]. Alternatively, one could work with quintic polynomials. The formulas for the different configurations and matches carry over and only the degree has to be raised. However, working entirely with quintics is costly and can be avoided if the design uses only isolated triangular patches.

Theorem 4.3 shows that fitting an irregular n-gon with quartic triangular patches as in [Shirman, Séquin 87] is only possible if the boundary curves are chosen such that $k_0 = k_1$. If the n-gon is regular, a regular 'star-like' construction for the tangents at the splitting point is natural and leads to $k_0 = k_1$. For irregular data this construction can be altered so as reflect the data while maintaining equality of the tangent ratios. If n and thus the splitting point is even, the quartic mesh curves of Section 6 can be used to ensure compatibility.

5. The algorithm: interpolation of a cubic mesh

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

step 1: [Determine tangent ratios] At each data point, solve

$$\eta_i u_i^0 = k_i v_i^0 + (1 - k_i) w_i^0 \qquad (E_0')$$

for k_i and η_i and set $\lambda_i^0 := \alpha_i \eta_i$, $\mu_i^0 := \alpha_i k_i$ and $\nu_i^0 := \alpha_i (1 - k_i)$.

Remarks:

(a) Assumption 1.c, which stipulates the proper distribution of the neighbors of any point, implies $0 < k < 1$. This guarantees that the match satisfies (I), and hence is *cusp-free*.

(b) A 3-3 configuration with $k_0 \neq k_1$ requires special treatment (see Section 4.3).

(c) The weights of a 3-4 configuration have to be properly scaled (see Section 4.2, Remark (b)). Otherwise $\alpha = 1$ is sufficient.

step 2: [Determine twists] If the number of incident edges is odd, solve (T) with λ_i^1 chosen by degree raising. Otherwise solve (T'). Since the corresponding system is underdetermined, we propose to enforce

$$\min \frac{1}{2} \sum_{i=1}^{n} (\|t_i^* - t_i\|^2 + \|\lambda_i^* - \lambda_i^1\|^2)$$

$$\text{s.t.} \quad \begin{pmatrix} K & 0 & 0 & U^x \\ 0 & K & 0 & U^y \\ 0 & 0 & K & U^z \end{pmatrix} \begin{pmatrix} T^x \\ T^y \\ T^z \\ L \end{pmatrix} = \begin{pmatrix} R'^x \\ R'^y \\ R'^z \end{pmatrix},$$

where t_i^* and λ_i^* are 'desirable' values for t_i and λ_i^1. In particular, we propose to use degree raising to obtain the λ_i^*, e.g. as $2\lambda(0) + \lambda(1)$ if λ is cubic, and, following [Sarraga '86], use blending for the estimate $t_i^* := P + u_i^0 + u_{i+1}^0$.

Remark: The least squares solution is also applied to the global boundary of an open surface where the constraints number one less than in the interior.

Figure 5: Coefficients determined after step 2

step 3: [Determine the interior coefficients of biquartics] For biquartic patches: compute the middle coefficients of the first layer using (9) or (11) as detailed in Sections 4.1 and 4.2.

step 4: [Determine the center coefficient of biquartics] For biquartic patches: determine the central coefficient as an average of the surrounding coefficients.
Remark: The central coefficient of biquartic patches can be chosen freely.

6. Generation of compatible meshes using quartic space curves

The goal of this section is to locally construct a mesh of piecewise quartic space curves consistent with given second order data at the mesh points. A similar construction can be found in [Peters 88c]. We do not repeat the match and configuration pairs of Sections 4.1 to 4.3, since it is straightforward to adapt the formulas for $\widetilde{u} = \{u^0, 3u^{01}, 3u^{10}, u^1\}$. (It took the author less than an hour to adapt the implementation and generate Figures 6(e) and 6(f).) Note, that due to higher degree of the curve mesh, we loose one scalar degree of freedom.

We start the construction of

$$p(u,0) =: \{P^0, 4B^0, 6C, 4B^1, P^1\}$$

by fitting a cubic to the C^1 data at the end points. In particular, computing d^i as the projection of $P^1 - P^0$ into the tangent plane at P^i, we obtain C by raising the degree of the cubic $\{P^0, 3(P^0 + \frac{1}{3}d^0), 3(P^1 - \frac{1}{3}d^1), P^1\}$:

$$C := \frac{1}{2}(P^0 + P^1) + \frac{1}{8}(d^0 - d^1).$$

With C thus fixed, we can match the second order information by setting $B^i := P^i + (-)^i \delta^i t^i$ where $t^i := d^i / \|d^i\|$. Dropping the superscript i for simplicity, we can compute δ to match κ, the normal curvature in the direction t:

$$\kappa = \frac{\|p' \times p''\|}{\|p'\|^3} = \frac{\|4\delta t \times 12(C - B - \delta t)\|}{4^3\delta^3\|t\|^3} = \frac{3\|t \times (C - P)\|}{4\delta^2}.$$

This gives

$$B^0 = P^0 + \sqrt{\frac{3\|t^0 \times (C - P^0)\|}{4\kappa^0}}\, t^0.$$

If the second fundamental form is specified by the principal direction ξ and the two principal curvatures $\kappa_{\min}$ and $\kappa_{\max}$, κ can be computed as

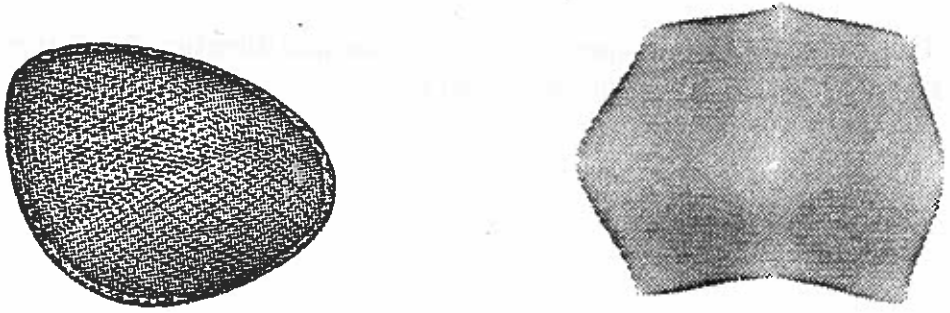
$$\kappa := \kappa_{\max}(\xi t)^2 + \kappa_{\min}((\xi \times N)t)^2$$

7. Examples

In this section, we display some of the surfaces generated by the algorithm. The first four surfaces interpolate cubic curve meshes, while the last two match quartic boundary curves. The cubic meshes are generated by prescribing the positions and normals of a set of data points and computing a curvature estimate. We then followed the approach of [Sabin '78] to obtain curves compatible with the curvature data whenever possible. The input of Figure 7(a), for example, consists of 4 points equally distributed on a sphere. The figure features six 3-3 configurations with $k_0 = k_1$. All points are 3-points. The 'cube' of Figure 7(b) illustrates 4-4 configurations. To obtain matches with $k_0 \neq k_1$ the upper right front point of the cube is been pulled up and towards the viewer. In both figures we display the 'z-buffered' and shaded surfaces overlaid by a B-net enlarged by 5%.

Figures 7(a-b): Interpolation of tetrahedral and (perturbed) cube data

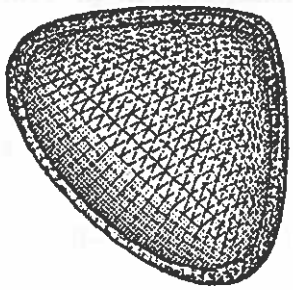
In Figure 7(c) we see the algorithm applied to mixed configurations. The mesh forms a prism: it bounds three rectangular and two triangular facets. Again the upper right front point has been pulled towards the viewer to obtain different types of matches. The four rectangular patches displayed in Figure 7(d) form a saddle (top view). Its nine data points can be easily identified by their white neighborhood, a result of the coincidence of light and normal directions. We used variations of the saddle data to test the implementation according to Theorem 3.2. The center point is a 4-point, and the setup in Figure 7(d) is chosen such that the mesh curves only conform to a second fundamental form but don't satisfy Sarraga's assumption. Figure 7(d) is also interesting in that it displays an open surface. The algorithm adapts in a straightforward fashion: the twist equations are solved via least squares with one fewer constraint.

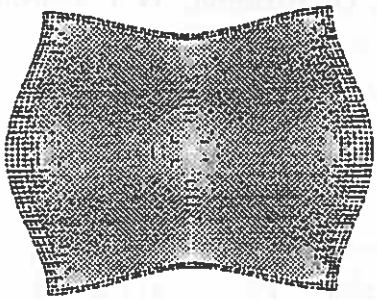


Figures 7(c-d): Prism and saddle

For comparison, we interpolate the same data first with a quartic curve mesh generated

according to Section 6. The quartic mesh resolves most of the problems, but the restrictions on 3-3 configurations suggest that splitting should be used in addition.





Figures 7(e-f): Prism and saddle over a quartic mesh.

Acknowledgement: I thank Carl de Boor for his comments and his help in proving Corollary 2.2.

References

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

de Boor, Carl (1987), B-form basics, *Geometric Modeling*, G. Farin ed., SIAM.

de Boor, Carl, Klaus Höllig, Malcolm Sabin (1987), *Computer Aided Geometric Design*, 4 pp 269-278

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

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

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. pp 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. pp 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

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

Hahn, Jörg (1987), Geometric Continuous Patch Complexes, *Technical Report*, Brunel University, presented at Oberwolfach 10 Feb 87, to appear in CAGD

Jones, A.K.(1988), NonrectaDangular surface patches with curvature continuity, *CAD*, 20, No. 6, pp 325-

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

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

Peters, Jörg (1988a), Local cubic and bicubic C^1 surface interpolation with linearly varying boundary normal, to appear in *Computer Aided Geometric Design.*

Peters, Jörg (1988b), Local piecewise cubic C^1 surface interpolants via splitting and averaging, submitted to *Computer Aided Design.*

Peters, Jörg (1988c), Local Generalized Hermite Interpolation by Quartic C^2 space curves, *Transactions on Computer Graphics*, July 1989.

Piper, B. (1986), Visually smooth interpolation with triangular Bézier patches, in *Geometric Modeling,* G.Farin, ed.

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

Sabin, Malcolm 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.*

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

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

Appendix: Savings for 'regular' 4-point geometries

In this section we focus on meshes or parts of meshes restricted to satisfy:

$$k_0 = k_1 \text{ and } \eta = 0.$$

The reference [Faux, Pratt '79, p.215] points out that $k_0 = k_1$ is a strong restriction on rectangular meshes [Faux, Pratt '79, p.215]. However, when triangular patches are included, this argument is less compelling. For example, we can design simple features like 'suitcase corners' in this setup. We consider the special setup since it offers considerable savings in storage and complexity. Since $k_0 = k_1$, the weight functions μ and ν may be constant polynomials. Further, we can limit ourselves to patches of degree three. Thus $\lambda = 0$ and the (1,0,0)-match for a cubic 4-4 configuration simplifies to

$$0 = \mu\{v^0, 3v^{01}, 3v^{10}, v^1\} + \nu\{w^0, 3w^{01}, 3w^{10}, w^1\}.$$

Once μ and ν are determined from the tangent constraints ($\alpha = 1$), we only need to solve

$$\mu v^{01} = -\nu w^{01}$$
$$\mu v^{10} = -\nu w^{10}.$$

Similarly, for 3-4 configurations, we have

$$0 = \{\mu, \mu\}\{v^0, 2\bar{v}, v^1\} + \nu\{w^0, 3w^{01}, 3w^{10}, w^1\},$$

and, for 3-3 configurations,

$$0 = \mu\{v^0, 2\bar{v}, v^1\} + \nu\{w^0, 2\bar{w}, w^1\}.$$

The inclusion of triangular patches leads to non-local systems of equations, since the compatibility conditions at adjacent data points share the center coefficient of the cubic. This non-localness is, however, bounded by any 'enclosure' of bicubic patches, since bicubics separate the compatibility conditions at adjacent points. Since $\lambda = 0$ everywhere, the compatibility constraints (T) are always solvable, and we have one twist coefficient per system free to choose. For example, we are entitled to choose the interior coefficient of the triangular patch in Figure 4.4 and can still solve (T) at the three corners of the patch.

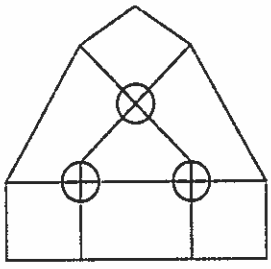


Figure A(a): An 'enclosed' triangle (schematic view)