Parametrizing Singularly to Enclose Data Points by a Smooth Parametric Surface

Jörg Peters
IBM T.J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598

Abstract

The problem of enclosing a data point by a smooth piecewise polynomial surface is solved with the help of a singular parametrization. The technique is illustrated by an algorithm that constructs a C^1 surface interpolating 3D positional and normal data.

Résumé

Une surface lisse á pièces polynomielles est construit autour d'un point par une parametrization singulaire. La technique est illustrée par un algorithme que construit une surface C^1 qu' interpole des dates 3D.

Key words: Bernstein-Bézier form, C^1 surface, interpolation, singular parametrization.

1. Introduction

Enclosing a data point as a vertex by a complex of patches is one of the more difficult tasks when constructing smooth surfaces. Intuitively, while the first $n-1$ patches around the vertex need only join across one edge, the last has to match across two edges. When all patches are to be determined simultaneously, the problem is a circular dependence among the smoothness constraints. In general, this dependence makes it impossible to construct a smooth, regularly parametrized interpolant (with one polynomial patch per mesh facet) to a given mesh of curves [Peters '89]. Even if no mesh of curves but just discrete data are prescribed, this implies that care has to be taken when generating the patch boundaries. Three techniques are known to overcome the problem. Gregory's rational patches [Gregory '74] break the dependence by allowing for a discontinuity in the second derivative: the first mixed derivative is not unique. A similar limitation of continuity within a mesh facet is achieved by splitting each patch into two and then joining the pair C^1 rather than C^2 (see e.g. [Farin '83]). A third technique is to force the boundary curves to match a second fundamental form at the vertex [Peters '89]. This is, for example, the case when four patches meet and the first and third, and the second and the fourth boundary curve join with high continuity (see e.g. [Bézier '77][Sarraga '86]). In particular, this explains why the vertex enclosure problem is not noticed for tensor-product constructions ([Coons '67], [Gordon '69]). On the other hand, it makes clear that the ease of such constructions is due to the special data and hence that it is difficult to extend them to the general case.

This paper offers a new, fourth alternative, namely singular (re)parametrization of the patches at the vertex. Rather than separating mixed derivatives, the idea is to make them trivially agree by setting the first derivative of each boundary curve and all mixed derivatives at the vertex to zero. By forcing the second boundary derivatives into a common plane, they take over the role of the first derivatives in defining the tangent plane. – In more detail, consider a parametrization p that maps the unit triangle or the unit square to $\mathbb{R}^3$. Denote the partial derivative in the direction of the i^{th} unit vector by D_i, abbreviate

$$p_{i\ldots j} := D_i \ldots D_j p(0,0)$$

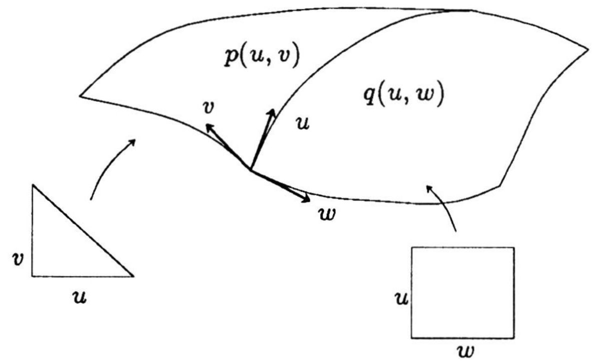
and let $P =: p(0,0)$ with N the normal at P. Then the construction *enforces*

$$p_1 = p_2 = p_{12} = 0 \quad \text{and} \quad p_{11} \perp N, p_{22} \perp N. \quad (1.1)$$

Note that while setting $p_{12} = 0$ induces a flat spot on the graph of a bivariate map into $\mathbb{R}$, this is in general not the case for a parametric surface (the target of a map from $\mathbb{R}^2$ to $\mathbb{R}^3$).

Section 2 reviews the vertex enclosure problem and proves the sufficiency of constraints 1.1. As an illustration, Section 3 gives a simple algorithm for the interpolation of a mesh of points and normals in 3-space.

2. The vertex enclosure problem



(2.1) Figure: Parametrization of abutting patches.

First-order continuity or *oriented tangent plane continuity* between two patches p and q can be characterized by the constraint

$$0 \stackrel{!}{=} f(u) := \lambda(u) D_1 p(u,0) - \mu(u) D_2 p(u,0) - \nu(u) D_2 q(u,0) \tag{2.2}$$

where λ, μ and ν are univariate scalar-valued functions such that $\mu\nu > 0$ (see e.g. [Liu '86 p.437], [Peters '88], [Degen '89 p.10], [Liu, Hoschek '89 Thm.1]). Setting $f(0) = 0$ for given p_1, p_2 and q_2 pins down $\lambda(0)$, $\mu(0)$ and $\nu(0)$ except for a common factor. Setting $f'(0) = 0$ for given $\lambda(0)$, $\mu(0)$, $\nu(0)$, p_1, p_2, q_2 and p_{11} leads to

$$\mu p_{12} + \nu q_{12} = \lambda p_{11} + (\lambda' p_1 - \mu' p_2 - \nu' q_2) \quad \text{at } 0. \tag{2.3}$$

For each of n boundary curves emanating from the vertex $P := p(0,0)$, there is one such constraint. Denote the i^{th} patch by p^i and let N be the normal at P. Since $N * (\lambda'(0) p^i_1 - \mu'(0) p^i_2 - \nu'(0) q^i_2) = 0$ and $p^i_{12} = p^i_{21}$, there is an $n \times n$ system of constraints

$$\mu^i N * p^i_{12} + \nu^i N * p^{i+1}_{12} = \lambda^i N * p^i_{11}. \quad \text{at } 0. \tag{2.4}$$

In general, for given p^i_{11}, this system is inhomogeneous and cannot always be solved since the constraint matrix is rank deficient whenever n is even (see e.g. [Watkins '88]).

Equation 2.4 can, however, be trivially enforced by using a singular parametrization, i.e. a parametrization p such that $\det Dp = 0$. This is unconventional from the point of view of differential geometry which emphasizes and often restricts itself to regular parametrizations ([do Carmo '76 p.52],[Klingenberg '83, 3.1.1]) since then 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 simple examples, e.g. $t \mapsto (t^2, t^3)$ 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.

(2.5) Lemma. *A vertex can be enclosed by a C^1 surface if the boundary curves emanating from it are parametrized singularly.*

Proof. Parametrize so that for $j \in \{0,1\}$ and q_i a tangent vector for the ith boundary curve,

$$D_j p^i|_0 = 0 \quad \text{and} \quad D_j^2 p^i|_0 = q_{i+j}. \tag{1.1'}$$

Then $f^i(0) = 0$ for any choice of $\lambda^i(0)$,$\mu^i(0)$ and $\nu^i(0)$, $\lambda^i(0) = 0$ and $p^i_{12} = 0$ imply $f'(0) = 0$ and the tangent plane is defined by the second derivative since $(\alpha D_1 + (1-\alpha) D_2)^2 p^i|_0 = \alpha^2 q_i + (1-\alpha)^2 q_{i+1}$. ♣

The algorithm in Section 3 constructs boundaries that satisfy (1.1) and shows that the technique, while not profound, is effective. The alternative approach, setting $N * p^i_{11} = 0$ to make (2.4) homogeneous, leads to flat spots if $p^i_1 \neq 0$.

3. An algorithm for example

The following algorithm represents bivariate vector-valued patches in Bernstein-Bézier-form (see e.g. [Farin '86], [de Boor '87]) to have easy access to value and derivative information along patch boundaries. In Figure 3.2 the natural association of coefficients with the unit domain is used to label the 21 coefficients of a quintic 3-sided (total-degree) patch. Enforcing $p_1 = p_2 = p_{12} = 0$ at P_i means setting

$$B^1_{ij} = B^1_{ik} = C_i = P_i.$$

That is, the algorithm below constructs 3-sided quintic patches with at most 12 distinct coefficients. The reader can check that a similar 4-sided biquartic (tensor-product) patch has 12+1 distinct coefficients (the center coefficient will be free to choose). Still the patches are underconstrained: the second difference vector $B^2_{ij} - B^1_{ij}$ needs only lie in the tangent plane. A smooth join across the edge ij is achieved by determining D_{ij}, resp. D_{ki}, appropriately. For no better reason other than the author's familiarity with the approach, smoothness across the boundaries is achieved by prescribing the boundary normal as a weighted linear blend of the normals at the end points ([Peters '88]). Thus, setting

$$n(u) := (1-u) N^0 + \omega\, u N^1,$$

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

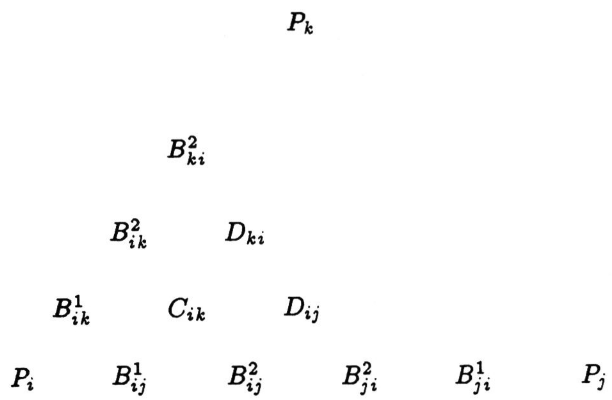
transforms the non-linear smoothness constraint

$$\det(D_1 p(u,0), D_2 p(u,0), D_2 q(u,0)) = 0 \qquad (3.1)$$

into the linear constraints

$$\begin{aligned} n(u) D_1 p(u,0) &= 0, && (e_u) \\ n(u) D_2 p(u,0) &= 0, && (e_v) \\ n(u) D_2 q(u,0) &= 0, && (e_w) \end{aligned}$$

for each edge and $u \in [0..1]$. (Constraint (3.1) is equivalent to (2.2).) We note that not all data allow for a linearly varying normal. For the present purpose, however, we refuse to worry and refer to [Peters '88] for solutions. Since (e_u), (e_v) and (e_w) are trivially enforced at $u = 0$ and $u = 1$, the only work consists of pinning down the underdetermined interior coefficients, D_{ij}. One option is to force the surface to be close to a cubic-biquadratic interpolant of the same data. However, this leads to 'bulgy' surfaces (cf. Figure 3.4). A better choice is to minimize the variation of the cross-boundary derivative and this is done below.



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

Algorithm

Input A mesh of data points and their normals such that $\omega > 0$. Each facet has 3 or 4 edges.
Output A quintic-biquartic C^1 surface that interpolates the mesh.

Tools $*$ is the vector product, $\times$ 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.
tanproj(v, n) – returns $(v - v * n)n$, the vector component of v perpendicular to n.

for i = 1:points [construct the boundaries B^l_{ij}]
  for m = 1:nbrs(i)
    l =nbr(i, m); [e.g. $l = j$ or $l = k$]

    [singularity at P_i]
    $B^1_{il} \leftarrow P_i$; $C_{il} \leftarrow P_i$;

    [construct the B^2_{il}]
    $\Delta = P_l - P_i$; $\sigma = N_l * \Delta$;
    if quintic **then**
      $t \leftarrow$ tanproj(Δ, N_i);
      $\alpha \leftarrow 3\sigma/(5t * N_l)$;
      if $\alpha \le 0$ [check for inflection]
        error [no linear $n(u)$ possible];
    else [biquartic patch]
      $m \leftarrow N_i \times N_l$;
      $t \leftarrow m \times N_i$;
      $\alpha \leftarrow \sigma/(m * m)$;
    endif
    $B^2_{il} \leftarrow P_i + \alpha t$;

for f = 1:faces [construct the D_{il}]
  for $i = 1 : \text{sds}(f)$ [$j = _{i+1}$ etc. below]
    $d_i = (B^2_{i,i-1} - P_i + B^2_{i+1,i+2} - B^2_{i+1,i})/2$;
    $F_i = (B^2_{i,i+1} + B^2_{i+1,i})/2$;
    $D^*_i = d_i + F_i$; [target value for $D_{i,i+1}$]
    solve

$$\begin{bmatrix} 1 & N_i N_{i+1} \\ N_i N_{i+1} & 1 \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} = \begin{bmatrix} (D^*_i - B^2_{i+1,i}) * N_i \\ (D^*_i - B^2_{i,i+1}) * N_{i+1} \end{bmatrix}; \qquad (3.3)$$

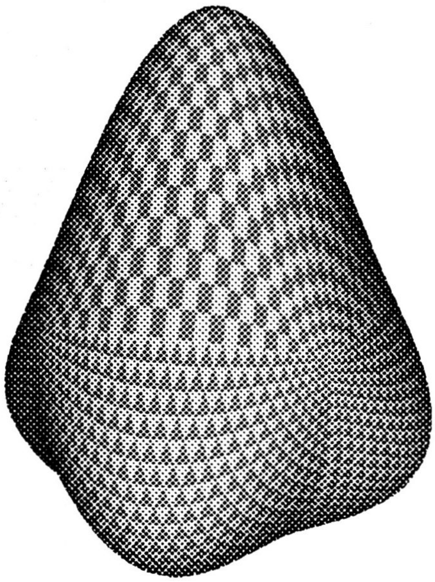
    $D_{i,i+1} \leftarrow D^*_i - l_1 N_i - l_2 N_{i+1}$;
  if sds(f) == 4
    biquartic center coefficient $\leftarrow$ average of the surrounding coefficients

The interpolants in Figures 3.4-6 fit to the same 5 vertices of an upside-down pyramid with square base. However, the first aims at minimizing the distance to a cubic-biquadratic interpolant, the second is the output of the above algorithm and the third is obtained by adjusting the tightness across one boundary by setting

$$D^*_i = d_i + F_i + \gamma N,$$

and increasing γ from 0 to 5. N is the average of the normals at the vertices of the patch. The figures show two of the four 3-sided and the 4-sided patch The checker pattern follows isoparametric lines and thus points to the singular vertices. Figure 3.7 shows the author's favorite object of genus 2 constructed with 50 patches. The effect of adjusting 'tightness' (altering the least squares target value) is illustrated in Figures 3.8 and 3.9.

Picking and deforming a patch is done in real time – the recomputation involves only one patch. Shading requires some care: since the parametric derivative is zero, the normal at a singular vertex cannot be computed from the parametrization. However, the normals are part of the input data and hence need not be recomputed. Adaptive subdivision does the rest.

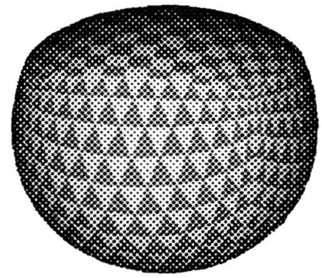


(3.4) Figure: Degree-minimizing interpolant to 5 points bounding an (inverted) pyramid. The square base gives rise to the 4-sided patch on top.

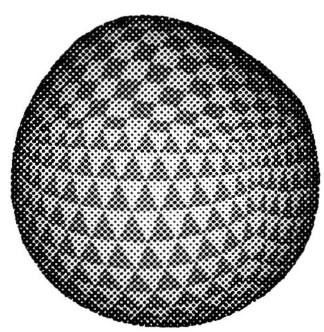
4.Conclusion

A new technique, singular polynomial parametrization at the vertices, is shown to overcome the problem of enclosing a vertex by a complex of patches. This means that one needs not resort to rational patches or change the topology by splitting patches in order to build smooth, interpolating surfaces of arbitrary genus. However, singular polynomial patches are of slightly higher degree and the calculation of normals close to the vertex requires care.

Acknowledgement: This research was supported by NSF DMS-8701275.



(3.5) Figure: Interpolant to 5 points minimizing the cross-boundary derivative (default: $\gamma = 0$).

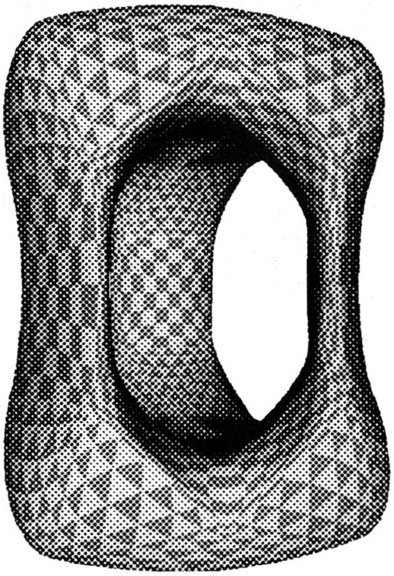


(3.6) Figure: Interpolant to 5 points with $\gamma = 5$.

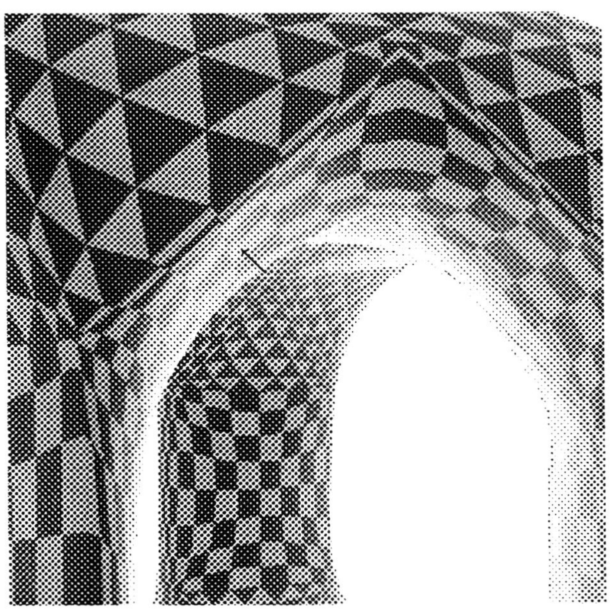
5. Appendix: Derivation of the algorithm

We first consider a quintic triangular patch and use the abbreviation

$$p \sim [b^0, \ldots, b^d] \quad \text{for} \quad p : t \mapsto \sum_{j=0}^{d} t^j (1-t)^{d-j} b^j,$$



(3.7) Figure: Genus 2 object (50 patches).



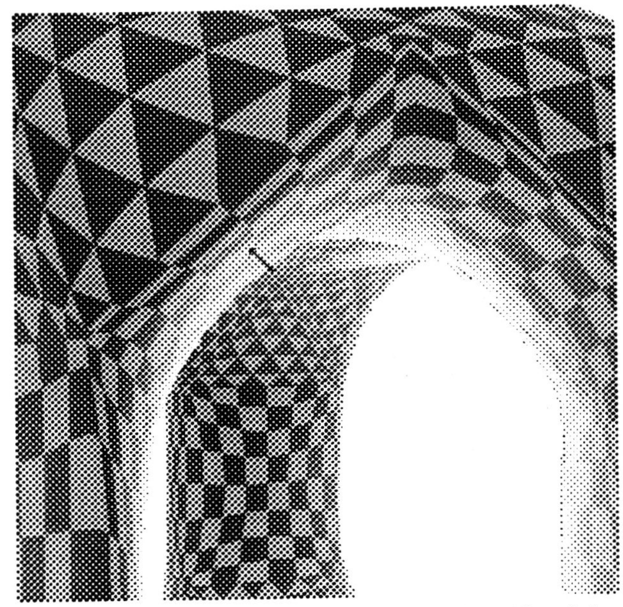
(3.8) Figure: Local adjustment of 'bulk'. Here $\gamma = 0$.

so that

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

where bc is the scalar product of b and c. Since

$$\begin{aligned} np_1 &\sim [n^0, n^1][u^0, 4u^1, 6u^2, 4u^3, u^4] \\ &:= [N_i, \omega N_j][B^1_{ij} - P_i, 4(B^2_{ij} - B^1_{ij}), 6(B^2_{ji} - B^2_{ij}), \\ &\quad 4(B^1_{ji} - B^2_{ji}), P_j - B^1_{ji}] \end{aligned}$$



(3.9) Figure: The 4-sided patch on the left side of the arch with $\gamma = 2$. (*nwarrow* is in the same position.)

is a polynomial of degree five and the setup is symmetric, it suffices to show that the first three coefficients of np_1,

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

are zero. By choosing $B^1_{ij} = P_i$ and forcing B^2_{ij} to lie in the tangent plane, e.g. $B^2_{ij} = P_i + \alpha t$ with t the projection of $P_j - P_i$ into the tangent plane at P_i, the first two coefficients vanish and the boundary curve is normal to N_i. It remains to show that

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

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

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

or, equivalently,

$$\begin{aligned} 5u^1n^1 &= 3\Delta(2n^0 + 3n^1) \\ 5u^3n^0 &= 3\Delta(3n^0 + 2n^1). \end{aligned}$$

Hence, the third coefficient vanishes if

$$\begin{aligned} \alpha &= \frac{3\Delta(2n^0 + 3n^1)}{5tn^1} = \frac{6\Delta N_i/\omega + 9\Delta N_j}{5tN_j} \\ &= \frac{-6\Delta N_j + 9\Delta N_j}{5tN_j} = \frac{3\sigma}{5tN_j}. \end{aligned}$$

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

$$np_1 \sim [n^0, n^1][u^0, 3u^1, 3u^2, u^3] =: [N_i, \omega N_j]$$
$$[B^1_{ij} - P_i, 3(B_{ij} - B^1_{ij}), 3(B^1_{ji} - B_{ij}), P_j - B^1_{ji}].$$

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:

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

The remaining term is zero by choice of ω:

$$\begin{aligned} n^0(B^1_{ji} - B_{ij}) + n^1(B_{ij} - B^1_{ij}) \\ &= (n^0 + n^1)(B^1_{ji} - B^1_{ij}) \\ &= (N_i + \omega N_j)\Delta = 0. \end{aligned}$$

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

$$N_i(B_{ij} - P_i) = 0 \quad \text{and} \quad N_j(B_{ij} - P_j) = 0.$$

The above version of the algorithm adds *adhoc* the constraint

$$(N_i \times N_j)(B_{ij} - P_i) = 0.$$

By setting $B_{i,l+1} = P_i$ and $C_{il} = P_i$, the first two coefficients of np_2 vanish. Noting that, in both the quintic and the biquartic case, np_2 is a quintic univariate polynomial, it suffices to enforce for each boundary in the quintic case

$$(D_{il} - B^2_{il})N_i = 0 \text{ and } (D_{il} - B^2_{li})N_l = 0, \qquad (5.1)$$

respectively in the biquartic case

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

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. A better alternative is to reduce the complexity of the transversal derivative by minimizing the difference between the central transversal difference, $D_{ij} - B^2_{ij}$, and the average of the first and last difference,

$$d_i := (B^2_{ik} - P_i + B^2_{jk} - B^2_{ji})/2.$$

Then (D_{ij}, D_{jk}, D_{ki}) are the solution of

$$\begin{aligned} &\min \|D_{ij} - D^*_i\|^2 + \|D_{jk} - D^*_j\|^2 + \|D_{ki} - D^*_k\|^2 \\ &\text{s.t.} \quad N_i D_{ij} = N_i B^2_{ij} \quad i,j \in \{1,2,3\}, i \neq j. \end{aligned} \qquad (5.2)$$

By duality, this is equivalent to (3.3). The deviation for the biquartic patch is 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.

References

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

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.

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

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.

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

Farin, Gerald (1986), Triangular Bernstein-Bézier patches, *CAGD* **3**.

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

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

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.

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

Peters, J. (1989), Smooth Interpolation of a Mesh of Curves, Constructive Approximation **7** (1991), 221-247.

Peters, J. (1990), Fitting smooth parametric surfaces to 3D data, CMS Tech. Report No. 91-2, U. of Wisconsin (1990).

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

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