ROCKY MOUNTAIN
JOURNAL OF MATHEMATICS
Volume 14, Number 1, Winter 1984

THREE- AND FOUR-DIMENSIONAL SURFACES

R.E. BARNHILL AND F.F. LITTLE

ABSTRACT. The representation and approximation of three- and four-dimensional surfaces is accomplished by means of local, piecewise defined, smooth interpolation methods. In order to interpolate to arbitrarily located data, the schemes are defined on geometric domains of triangles or tetrahedra, respectively.

Introduction. This research creates new methods for representing and approximating three and four-dimensional surfaces. The numerous applications of surface methods include modelling physical phenomena (e.g., combustion) and designing objects (e.g., airplanes and cars). In addition to three-dimensional surfaces, there are interesting four-dimensional "surfaces", such as temperature as a function of the three spatial variables. Because the geometric information for these problems can be located arbitrarily in three or four-dimensional space, the surface schemes must be able to handle arbitrarily located data. The standard (and easier) approach to surfaces of using tensor products of curve methods restricts the surface method's applicability to (rectangularly) "gridded" data. We take the more ambitious approach of devising robust surface methods applicable to arbitrarily located data. There are two broad classes of methods suitable for solving these problems (i.e., problems for which simplifying geometric assumptions cannot be made):

(1) local triangular patch methods, and
(2) global convex combination methods.

"Local" means that the value of the surface at a point is affected only by nearby data, whereas "global" means the opposite. In this paper we discuss "patch" methods only. (Global methods are discussed by Barnhill, Dube, and Little [4] and by Barnhill and Stead [6].)

Historically, surface methods have been built up from curve methods (via tensor products). We discuss surface methods that are truly multivariate; in particular, bivariate for 3D surfaces and trivariate for 4D surfaces. The domain of definition of our 3D surfaces is the union of

This research was supported in part by the National Science Foundation with Grant MCS-8101854 and by the Department of Energy with Contract No. DE-AC02-82ER 12046.A000 to the University of Utah.

triangles and of our 4D surfaces the union of tetrahedra, respectively. These geometric domains allow us to define surface interpolants for arbitrarily located points. Our surface interpolants are instances of "patch" methods, in which (small) curved pieces are joined together to form a smooth surface. Patch methods originated with S.A. Coons [7] and have been generalized considerably. (See [**1**, **2**].)

Patch methods on arbitrarily located data are also called piecewise triangular methods. The domain of the surfacing problem is decomposed into simplices and an interpolant is constructed on each of these. The composite surface is the triangular interpolant defined on the union of the simplices.

The continuity of the composite surface is used to label the simplicial interpolants; so a C^1 triangular interpolant is one which gives a C^1 composite surface when it is used on neighboring simplices. The continuity of interpolants is generally a consequence of the particular interpolation. The interior continuity of interpolants is not usually an issue; so the continuity of the composite surface is determined by values and derivatives matching up at interfaces.

There are two pre-processing steps in the definition of our (C^1 smooth) surface interpolants:

(1) define the geometric domain, and

(2) create gradients.

In step (2) we are using the fact that our C^1 surfaces require only C^1 data. If the user supplies only C^0 data, then the C^1 data must be created. Each of pre-processing steps (1) and (2) is challenging and interesting; we present one suite of solutions. Users ordinarily want smoother surfaces than their data imply directly, so that additional information must sometimes be created. (A notable feature of our methods is that the smoothness of the surface is always greater than or equal to the smoothness of the data. This is not true of most other schemes such as finite element schemes.)

We present two C^1 3D surface interpolants:

(i) a Barnhill, Birkhoff, and Gordon (BBG) scheme, and

(ii) a Radial Nielson scheme.

These are "transfinite" interpolants, which means that whole curves of positional and derivative information are interpolated. Then we discretize these transfinite interpolants to obtain finite-dimensional patches, i.e., patches which depend on only finitely many data points. Transfinite interpolants have a unified theory which makes it relatively easy to trace the mathematical properties of their discretizations. The discretizations of the BBG and the Nielson schemes are both nine-degrees-of-freedom C^1 triangular interpolants. They are distinct interpolants. (The discretized Nielson scheme turns out to be the same as Little's "Madison triangle", a nine-degrees-of-freedom scheme created per se.)

We create one 4D surface interpolant: a C^1 BBG tetrahedral interpolant, which is then discretized to a 28-degrees-of-freedom C^1 scheme.

1. Surface preprocessors.

A. Simplices and barycentric coordinates. The usual $\mathbf{R}^n$ cartesian coordinate system may be augmented by local coordinate systems. A simplex in $\mathbf{R}^n$ is the ordered collection $\{V_1, V_2, V_3, \ldots, V_n, V_{n+1}\}$ of points called the vertices of the simplex. A convex combination of $n + 1$ points is $\sum_{i=1}^{n+1} b_i V_i$ where $b_i \geqq 0$ and $\sum_{i=1}^{n+1} b_i = 1$. All points inside the simplex are convex combinations of the vertices. The combination coefficients for a given point V inside a simplex are found by solving the linear system:

$$V = \sum_{i=1}^{n+1} b_i V_i, \qquad 1 = \sum_{i=1}^{n+1} b_i.$$

Using each V as a column gives the matrix equation

$$\begin{bmatrix} V_1 & V_2 & \cdots & V_n & V_{n+1} \\ 1 & 1 & & 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ b_{n+1} \end{bmatrix} = \begin{bmatrix} V \\ 1 \end{bmatrix}. \tag{1.1}$$

The restriction that all $b_i \geqq 0$ limits V to the inside of the simplex. Removing it allows the b_i to be determined uniquely for any point in $\mathbf{R}^n$. The b_i are called the *barycentric coordinates* of V. The condition that the matrix is invertible is the same as its determinant being not zero. In $\mathbf{R}^2$ this is the statement that the three vertices are not collinear, and in $\mathbf{R}^3$ that the four vertices are not coplanar, etc. In other words, the simplex cannot be degenerate (lower dimensional). In $\mathbf{R}^2$ if the matrix equation for the barycentric coordinates is solved by Cramer's rule, we have

$$b_1 = \frac{\begin{vmatrix} V & V_2 & V_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} V_1 & V_2 & V_3 \\ 1 & 1 & 1 \end{vmatrix}}, \qquad b_2 = \frac{\begin{vmatrix} V_1 & V & V_3 \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} V_1 & V_2 & V_3 \\ 1 & 1 & 1 \end{vmatrix}} \cdots. \tag{1.2}$$

The determinant in the denominator is twice the area of the triangle oriented counterclockwise. The determinant in each numerator is also the area of a subtriangle giving each b_i as the ratio of areas or masses, $b_i = A_i/A$.

B. Triangulation.

(1) Triangulation in two dimensions. Suppose that we have the positions $\{(x_i, y_i, z_i)\}_{i=1}^n$ and that we are considering a (nonparametric) surface of the form $z = F(x, y)$. (The latter assumption implies that the data are meaningful for defining such a surface.) Therefore, we consider triangulating the (planar) points $\{(x_i, y_i)\}_{i=1}^n$. Triangulations are not unique; so we have the following pair of questions.

(1) How do we find a triangulation?

(2) How do we measure the quality of various triangulations?

(We assigned the problem of triangulation as a class problem in 1976. We were amazed to find that it was not only an open problem, but a relatively unexplored problem.) The first step of finding a triangulation is discussed in **[1]**. The second step of "optimization" of a triangulation has the following history. Lawson **[12]** had been developing the idea of local optimization by switching the diagonals of convex quadrilaterals formed by two triangles. Then Green and Sibson **[8]** and Sibson **[17]** published papers on (Delaunay) triangulations obtained via Thiessen polygons. (Thiessen polygons arise often in nature; given $\{(x_i, y_i)\}_{i=1}^n$, partition the plane into "tiles" of influence such that the i-th tile is the set of points closest to (x_i, y_i).) Sibson proved the equivalence of Delaunay triangulations and Lawson's "locally equiangular triangulations". Lawson's optimization criterion is the following:

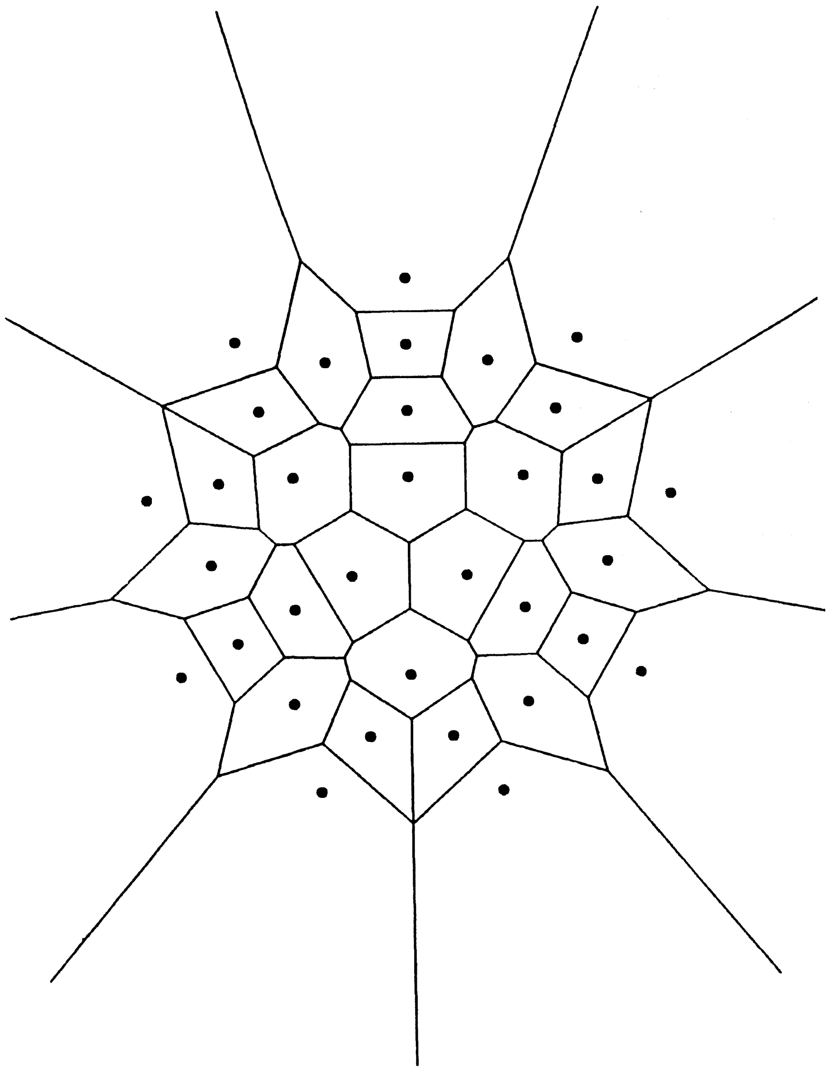
$$(1.3) \qquad \max_{\text{all triangulations } T} \quad \min_{\text{triangles in } T} \text{angle}$$

The idea is to keep the smallest angle in successive triangulations from approaching zero. A theoretical justification of this criterion is that a (majorizing) error bound contains the factor $1/\sin^n\theta$. However, Gregory **[11]** showed that the theoretical error bounds could be sharpened so that the angles only need be kept from approaching π. Thus we now recommend the optimization criterion:

$$(1.4) \qquad \min_{\text{all } T} \quad \max_{\text{triangles in } T} \theta$$

There is one theoretical shortcoming of our recommended optimization; Lawson's equiangular triangulation is a local method because only pairs of triangles are considered at a time. Sibson's Delaunay triangulation is global because the underlying Thiessen polygonalization is formed by considering all the points, i.e., a new point is (theoretically) considered relative to all the other points. Since Lawson's and Sibson's triangulations are equivalent, Lawson's is therefore a global optimum. We cannot make such a claim about our recommended criterion.

We conclude our planar triangulation discussion with a computational note. For computational simplicity we replace θ by $\cot\theta$ in criterion (1.4.). Since $\cot\theta$ is a monotone decreasing function, this replacement leads to the optimization criterion:



Thiessen polygonalization

$$(1.5) \qquad \underset{\text{all } T}{\text{maximize}} \quad \underset{\text{all triangles in } T}{\text{minimum}} \quad \cot\theta$$

$$\text{with } \cot\theta = \frac{\cos\theta}{\sin\theta} = \frac{\dfrac{e_1 \cdot e_2}{|e_1|\,|e_2|}}{\dfrac{2\text{ area}}{|e_1|\,|e_2|}} = \frac{e_1 \cdot e_2}{e_1 \cdot e_2^{\perp}}$$

where θ has adjacent sides e_1 and e_2 and $(a, b)^{\perp} \equiv (b, -a)$.

A stopping criterion for forming a triangulation can be in terms either of edges or vertices, since T triangles can be written as $T = (e_b + 2e_i)/3 = V_b + 2V_i - 2$ where e_b and e_i are the numbers of boundary edges and interior edges and V_b and V_i are the numbers of boundary vertices and interior vertices, respectively. (See **[8]**.)

There are many additional issues on triangulation, including how to store the triangles efficiently and how to evaluate the triangular interpolant.

(2) Triangulation in n dimensions.

(i) Initial triangulation. As we saw above, a triangulation of a planar point set can be constructed as the dual to the Thiessen polygonalization. A triangulation should subsequently be optimized. The following algorithm builds this polygonalization. Put the first point into a region with no boundaries. For each sequential point, find the region it is in and carve out a new region for it from the former region and its neighbors. Details of data structure and refinements of the algorithm follow. An n-dimensional point is stored as a real n-vector. A region associated with each data point is stored as a list of boundaries. A boundary is an affine equation which separates $\mathbf{R}^n$ into a positive half and a negative half. This is stored as an $n + 1$ vector and a back pointer to two bounded regions. Also describing this boundary is the point set common to both regions. This point set or explicit boundary can be infinite in extent. The explicit boundary is described as the projective image of the convex span of a set of points in projective n-space P^n, that is, the boundary set is all points in $\mathbf{R}^n$ of the form

$$\sum_{i=1}^{m} a_i d_i V_i \Big/ \sum_{i=1}^{m} a_i d_i$$

where $V_i \in \mathbf{R}^n$ is the projective image of

$$\left\{ \frac{d_i V_i}{d_i} \right\}$$

in P^{n+1}. A point of infinity in the direction P is stored as $\left\{ \frac{P}{0} \right\}$.

Given this data structure the Thiessen region containing a point (not in the current data point set) is found by starting in any region and checking that it lies on the proper side of all its boundaries. If it fails, one starts over on the region on the other side of the culprit boundary.

The carving out process for a point P proceeds as follows. Make a set L of regions (for consideration) initially consisting of the Thiessen region of the first point. Until L is empty, extract a defining point $P_{\prime}$ from L. Construct the implicit boundary $(P_{\prime} - P)\cdot V - (P_{\prime} - P)\cdot$

$(P_{\prime} + P)/2$; include it in the boundary for $P_{\prime}$ and P and construct the back pointers. Trim the existing explicit boundaries of $P_{\prime}$ to be on the positive side of the new boundary and, if any become empty, discard them. If a boundary was trimmed or discarded, then include the region on the other side of it from $P_{\prime}$ in L. Construct an unbounded explicit representation for the new boundary and trim it with all the existing implicit boundaries of $P_{\prime}$.

The unbounded explicit representation for a new boundary between P and $P_{\prime}$ is the collection of points

$$\left\{\frac{(P + P_{\prime})/2}{1}\right\} \text{ and } \left\{\frac{\pm E_i}{0}\right\}$$

where E_i are the second-through-n-th vectors resulting from the Gram-Schmidt orthogonalization of $\{P_{\prime} - P, e_1, \ldots, e_n\}$, where $e_1, \ldots, e_n$ are the usual unit coordinate vectors, These $n - 1$ vectors are orthogonal to $P_{\prime} - P$ as well as to each other and so they describe the boundary of the Thiessen polygonalization, These explicit boundaries are trimmed against implicit boundaries or affine equations by clipping each segment between pairs of elements against the equation to produce a set of "new" points, The equation is zero for new points This set of new points is reduced to its convex hull via a simple $(n - 2) - d$ convex hull algorithm, The "new" points and those old points on the "keep" side of the equation are reduced to their convex hull via the same general brute force algorithm.

"Clipping" is a concept from computer graphics, Given a line-segment between two projective points and an affine equation which changes sign on the segment, the problem is to find the point on the segment where the equation is zero, and so describe the positive and negative subsegment,

Let P_1 and P_2 be the projectively represented points (i.e., $n + 1$ tuples) and let $A(P) = A \cdot P$ where A is an $(n + 1)$-tuple and P is projectively represented The zero point is

$$P_0 = P_1 - (P_2 - P_1)A(P_1)/(A(P_2) - A(P_1)),$$

If $P_1 = P_2$, P_0 is the $(n + 1)$-tuple of zeroes which is outside P^n and hence is disregarded, A simple convex hull algorithm is available from our previous discussion of barycentric coordinates, One takes a point set and considers all possible simplices which can be constructed from it, One then culls out all points whose barycentric coordinates are positive relative to the simplex The resulting point set is the convex hull of the original,

(ii) Optimal triangulation. The number of tetrahedra T can be written as $T = (F_b + 2F_i)/4 = (1/3)E_b + E_i - V_i - 1$, where F_b and F_i are the number of boundary faces and interior faces and E_b, E_i, V_i are the number of boundary edges, interior edges, and interior vertices, respectively, (See **[8]**.)

Proceed as in two dimensions. A local region is identified. All possible triangulations are enumerated and the best one is chosen. In two dimensions the region was a convex quadrilateral identified with two triangles sharing an edge. The choice of best triangulation may insert the other diagonal. In three dimensions two tetrahedra may share a face and the five-vertex convex polytope has the alternative triangulation consisting of three-tetrahedra pairwise sharing faces containing a common "polar" edge. Also worth considering in three dimensions is the case where four tetrahedra share a common edge form an octahedron. Three alternative triangulations exist for the octahedron identified by the choice of a diagonal as the common edge of the four tetrahedra.

The octahedral optimality via a sequence of optimization steps using the former swap which always improved the triangulation. Surely all counter examples are not eliminated with this one so the importance of a good initial triangulation is paramount.

Closely related to the goal of minimizing the maximum angle in a triangle is the goal of maximizing the minimum altitude of a triangle. This goal has an $n-d$ generalization and was used in the tetrahedral case. We also used: minimize over all triangulations the maximum over all tetrahedra the quotient of the radii of the inscribed and circumscribed spheres. This criterion is used in the pictures. The pictures illustrate the optimization procedure for the given five points. We make an initial triangulation into three tetrahedra, which are rendered with the three tetrahedra together and then slightly separated. The optimal triangulation yields (in this example) two tetrahedra, which are analogously rendered. (We thank Paul Arner for these pictures.) For additional details and examples on optimal tetrahedra, see **[16]**.

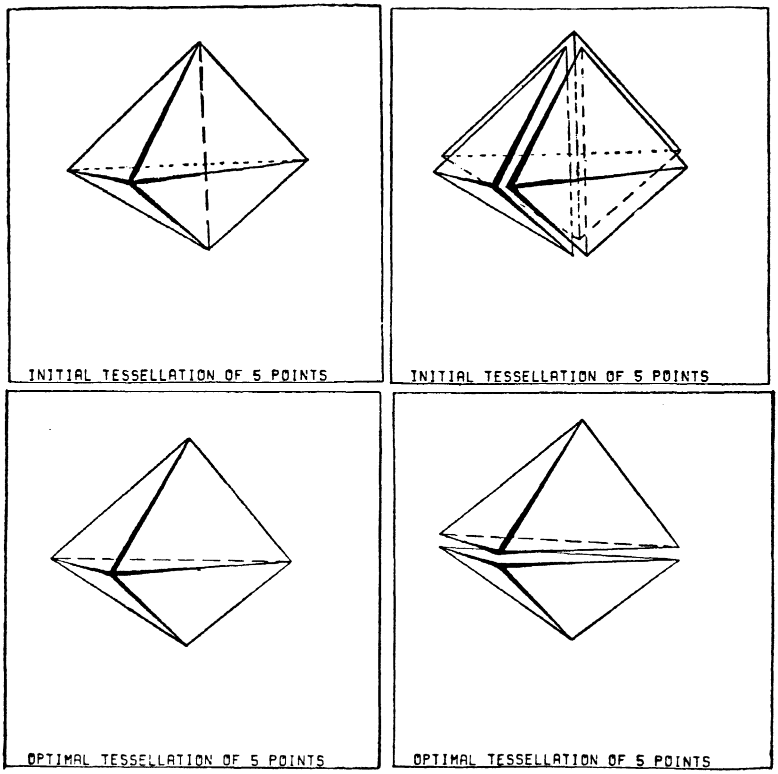
C. Gradients' Estimation. As noted earlier, gradients are needed to provide C^1 data for our C^1 surfaces. We recommend two methods for creating gradients:

(i) Triangular Shepard's Method, or

(ii) Inverse-distance-weighted Polynomial Least Squares constrained to interpolate.

Little **[14]** points out that the Triangular Shepard's Method is a global method whose gradients at the data points are local. (Triangular Shepard's Method to find gradients is a component of the surface package described in the PLOT 79 Users Manual available from N. L. Beebe, College of Science, University of Utah, Salt Lake City, Utah 84112.) Both (i) and (ii) above have n-dimensional analogues.

The n-dimensional Triangular Shepard is the following method. Let



T be a triangulation in n dimensions of a point set $\{\mathbf{x}_i,\ i = 1, \ldots, m\}$. Define

$$L(F;\, \mathbf{x}) \equiv \sum_{s \in T} L_s(\mathbf{x}) \cdot W_s(\mathbf{z})$$

where

$$W_s(\mathbf{x}) = \frac{1}{\prod\limits_{\mathbf{x}_i \in s} d_i^2(\mathbf{x})} \Bigg/ \sum_{s \in T} \frac{1}{\prod\limits_{\mathbf{x}_i \in s} d_i^2(\mathbf{x})}$$

with $d_i^2(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|^2$, and $L_s(\mathbf{x})$ is the linear interpolant

$$L_s(\mathbf{x}) = \sum_{\mathbf{x}_i \in s} b_i(\mathbf{x}) F(\mathbf{x}_i).$$

Then

$$\nabla L(F;\, \mathbf{x}_i) = \sum_{\substack{s \in T \\ \mathbf{x}_i \in s}} \nabla L_s \cdot U_{s,i}$$

where

$$U_{s,i} = \prod_{\substack{\mathbf{x}_j \in s \\ j \neq i}} \frac{1}{d_j^2(\mathbf{x}_i)} \Big/ \sum_{\substack{s \in T \\ \mathbf{x}_i \in s}} \frac{1}{\prod_{\substack{\mathbf{x}_j \in s \\ j \neq i}} d_j^2(\mathbf{x}_i)} .$$

(For more details see [14].)

2. Surface interpolants.

A. Barnhill, Birkhoff, and Gordon interpolants. Two popular bivariate surface interpolants are C^1 Clough-Tocher (see [13]) and Little's "Madison triangle," neither of which has a known trivariate analogue. However, the discretized Barnhill, Birkhoff, and Gordon (BBG) [3] interpolant does have a trivariate analogue. We develop a general form of the bivariate C^1 BBG interpolant in subsection (1) and extend it to the trivariate case in subsection (2).

(1) Three-dimensional BBG schemes.

(i) Transfinite. Univariate cubic Hermite interpolation to $\hat{f} = \hat{f}(t)$ defined on $[0, 1]$ can be written as

$$H_3(\hat{f}; t) = h_0(t)\hat{f}(0) + h_1(t)\hat{f}(1) + \bar{h}_0(t)\hat{f}'(0) + \bar{h}_1(t)\hat{f}'(1) \tag{2.1}$$

where $h_0(t) = 1 - 3t^2 + 2t^3$, $h_1(t) = 3t^2 - 2t^3$, $\bar{h}_0(t) = t - 2t^2 + t^3$ and $\bar{h}_1(t) = -t^2 + t^3$ for $0 \leqq t \leqq 1$. For $f = f(x)$ defined on $[a, b]$, we have

$$\begin{aligned} H_3(f; x) = {} & h_0\left(\frac{x-a}{b-a}\right)f(a) + h_1\left(\frac{x-a}{b-a}\right)f(b) \\ & + \bar{h}_0\left(\frac{x-a}{b-a}\right)(b-a)f'(a) + \bar{h}_1\left(\frac{x-a}{b-a}\right)b-a)f'(b). \end{aligned} \tag{2.2}$$

If F is a function on $\mathbf{R}^n$ and A and B are points in $\mathbf{R}^n$, then the cubic Hermite interpolant can be defined on $\overline{AB}$ by the analogous change of variables as

$$\hat{H}_3(F; X) = h_0(t)F(A) + h_1(t)F(B) + \bar{h}_0(t)\frac{\partial F(A)}{\partial(B-A)} + \bar{h}_1(t)\frac{\partial F(B)}{\partial(B-A)}$$

where $X = (1 - t)A + tB$ and

$$\frac{\partial F(V)}{\partial(B-A)} \equiv \frac{d}{dt}F(V + t(B - A))\Big|_{t=0} \tag{2.3}$$

is the Gateaux derivative of F at V. The BBG Projector is defined by cubic Hermite interpolation parallel to an edge of a simplex.

An edge of a simplex is the intersection of $n - 1$ faces. Each face is the zero contour of one of the barycentric coordinates; so a parallel to an edge can be described as the intersection of the contours of $n - 1$ barycentric coordinates. The segment endpoints for BBG interpolation are the intersections of this parallel and the two remaining faces of the simplex. We specialize these n-dimensional remarks to $n = 2$. Since

$$\begin{aligned} V &= b_1V_1 + b_2V_2 + b_3V_3 \\ &= \left(1 - \frac{b_3}{b_2 + b_3}\right)(b_1V_1 + (b_2 + b_3)V_2) + \frac{b_3}{b_2+b_3}b_1V_1 + (b_2 + b_3)V_3) \\ &(= (1 - t)A + tB) \end{aligned}$$

the BBG "lofting" interpolant parallel to the edge $\{b_1 = 0\}$ is

$$\begin{aligned} P_1(F; b_1V_1 + b_2V_2 + b_3V_3) &= h_0\left(\frac{b_3}{b_2 + b_3}\right)F(b_1V_1 + (b_2 + b_3)V_2) \\ &+ h_1\left(\frac{b_3}{b_2 + b_3}\right)F(b_1V_1 + b_2 + b_3)V_3) \\ &+ \bar{h}_0\left(\frac{b_3}{b_2 + b_3}\right)(b_2 + b_3)\frac{\partial F(b_1V_1 + (b_2 + b_3)V_2)}{\partial(V_3 - V_2)} \\ &+ \bar{h}_1\left(\frac{b_3}{b_2 + b_3}\right)(b_2 + b_3)\frac{\partial F(b_1V_1 + (b_2 + b_3)V_3)}{\partial(V_3 - V_2)}. \end{aligned} \tag{2.4}$$

The other two BBG lofting interpolants are defined analogously. Interpolant P_1 matches F and F's partial derivative in the direction $(V_3 - V_2)$ on $\overline{V_1V_2}$ and $\overline{V_1V_3}$. If F has a derivative in the direction of an interpolated edge, then P_1 also interpolates that derivative. (We only assume that the linear functionals which appear in the interpolant are defined. The function F is not assumed to be C^1, C^2 or even C^0.)

The interpolation properties of BBG Projectors can be combined via the Boolean sum. For P and Q linear interpolation operators, $P \oplus Q = P + Q - PQ$. Formally $P \oplus Q = Q + P(I - Q)$ is a correction surface form and has a permanence principle. Barnhill and Gregory's Theorem [5] states this as $P \oplus Q$ has at least the interpolation properties of P and the function precision of Q.

THEOREM. *The Boolean sum $A \oplus B$ has at least the function precision of B and the interpolation properties of A.*

PROOF. Let B be precise for F, i.e., $BF \equiv F$. Then $(A \oplus B)F = AF + BF - A(BF) = AF + F - AF = F$; $A \oplus B$ is precise for F.

Let L be an interpolation functional of A, i.e., L is a linear functional for which $LAF = LF$ for all relevant F. Then $L(A \oplus B)F = LAF + LBF - (LA)BF = LF$.

The triple Boolean sum $P_3 \oplus P_2 \oplus P_1$ interpolates the value and some cross-boundary derivative on all three sides of a triangle. If F has a gradient on the boundary of the triangle, this interpolant will match it. The goal of this section is to construct the triple Boolean sum $P_3 \oplus P_2 \oplus P_1$.

If all 124 terms of the complete expansion are written out, all but 15 cancel. This behavior can be explained at a high level as follows. Rewrite the triple Boolean sum as

$$P_3 \oplus P_2 \oplus P_1 = P_1 + P_2(I - P_1) + P_3(I - P_2)(I - P_1).$$

Since P_2 and P_1 both interpolate to the side opposite vertex three, we expect cancellation of those terms of $P_2(I - P_1)$ which involve evaluations on this side. In order to be explicit we introduce the additional notation $P_2 = P_2^1 + P_2^3$ where the superscript denotes the side on which the interpolant terms are evaluated. On the side opposite vertex three, $(I - P_1)F \equiv 0$ for all F and $\partial(I - P_1)F/\partial(V_1 - V_3) \equiv 0$. This causes $P_2^3(I - P_1)F \equiv 0$ for all F with a gradient.

This statement taken as a definition extends the domain of the interpolant. Similarly $P_3^1(I - P_2)F \equiv 0$ for all F and $P_3^2(I - P_1)F \equiv 0$. These observations lead to the simplification

$$P_3 \oplus P_2 \oplus P_1 = P_1 + P_2^1(I - P_1) - P_3^2 P_2^1(I - P_1). \tag{2.5}$$

We now enumerate the terms of this interpolant. In order to evaluate $P_2^1 P_1 F$, we calculate $P_1 F$ and $\partial P_1 F/\partial(V_1 - V_3)$ at an arbitrary point of edge one:

$$\begin{aligned} P_1(F; tV_2 + (1 - t)V_3) &= h_0(1 - t)F(V_2) + h_1(1 - t)F(V_3) \\ &+ \bar{h}_0(1 - t)\frac{\partial F(V_2)}{\partial(V_3 - V_2)} + \bar{h}_1(1 - t)\frac{\partial F(V_3)}{\partial(V_3 - V_2)}, \end{aligned}$$

and

$$\begin{aligned} \left.\frac{\partial P_1 F}{\partial(V_1 - V_3)}\right|_{tV_2+(1-t)V_3} &= -t\Big[h_0'(1 - t)F(V_2) + h_1'(1 - t)F(V_3) \\ &+ \bar{h}_0'(1 - t)\frac{\partial F(V_2)}{\partial(V_3 - V_2} + \bar{h}_1'(1 - t)\frac{\partial F(V_3)}{\partial(V_3 - V_2)}\Big] \\ &+ h_0(1 - t)\frac{\partial F(V_2)}{\partial(V_1 - V_2)} + h_1(1 - t)\frac{\partial F(V_3)}{\partial(V_1 - V_3)} \\ &+ \bar{h}_0(1 - t)\left[\frac{\partial^2 F(V_2)}{\partial(V_1 - V_2)\partial(V_3 - V_2)} - \frac{\partial F(V_2)}{\partial(V_3 - V_2)}\right] \\ &+ \bar{h}_1(1 - t)\left[\frac{\partial^2 F(V_2)}{\partial(V_1 - V_3)\partial(V_3 - V_2)} - \frac{\partial F(V_3)}{\partial(V_3 - V_2)}\right]. \end{aligned}$$

Hence

$$
\begin{aligned}
P_2^1(I-P_1)F = {} & h_0\left(\frac{b_1}{1-b_2}\right)\Bigg[F(b_2V_2+(1-b_2)V_3)-h_0(1-b_2)F(V_2) \\
& -h_1(1-b_2)F(V_3)-\bar{h}_0(1-b_2)\frac{\partial F(V_2)}{\partial(V_3-V_2)}-\bar{h}_1(1-b_2)\frac{\partial F(V_3)}{\partial(V_3-V_2)}\Bigg] \\
& +\bar{h}_0\left(\frac{b_1}{1-b_2}\right)(1-b_2)\Bigg[\frac{\partial F(b_2V_2+(1-b_2)V_3)}{\partial(V_1-V_3)}+b_2\Bigg\{h_0'(1-b_2)F(V_2) \\
& +h_1'(1-b_2)F(V_3)+\bar{h}_0'(1-b_2)\frac{\partial F(V_2)}{\partial(V_3-V_2)}+\bar{h}_1'(1-b_2)\frac{\partial F(V_3)}{\partial(V_3-V_2)}\Bigg\} \\
& -h_0(1-b_2)\frac{\partial F(V_2)}{\partial(V_1-V_2)}-h_1(1-b_2)\frac{\partial F(V_3)}{\partial(V_1-V_3)} \\
& -\bar{h}_0(1-b_2)\left[\frac{\partial^2F(V_2)}{\partial(V_1-V_2)\partial(V_3-V_2)}-\frac{\partial F(V_2)}{\partial(V_3-V_2)}\right] \\
& -\bar{h}_1(1-b_2)\left[\frac{\partial^2F(V_3)}{\partial(V_1-V_3)\partial(V_3-V_2)}-\frac{\partial F(V_3)}{\partial(V_3-V_2)}\right]\Bigg].
\end{aligned}
\tag{2.6}
$$

For the triple Boolean sum's last term, $-P_3^2P_2^1(I-P_1)$, we follow a similar course. Interpolant P_3^2 contains two functionals, namely, evaluation at $SV_3+(1-S)V_1$ and $\partial/\partial(V_2-V_1)|_{SV_3+(1-S)V_1}$. First,

$$
\begin{aligned}
& P_2^1(I-P_1)(F;\, SV_3+(1-S)V_1) \\
&= h_0(1-S)[F(V_3)-F(V_3)]+\bar{h}_0(1-S)\left[\frac{\partial F(V_3)}{\partial(V_1-V_3)}-\frac{\partial F(V_3)}{\partial(V_1-V_3)}\right] \equiv 0.
\end{aligned}
$$

Next

$$
\begin{aligned}
\left.\frac{\partial P_2^1(I-P_1)F}{\partial(V_2-V_1)}\right|_{SV_3+(1-S)V_1} &= h_0(1-S)\left[\frac{\partial F(V_3)}{(V_2-V_3)}+\frac{\partial F(V_3)}{\partial(V_3-V_2)}\right] \\
&\quad +\bar{h}_0(1-S)\Bigg[\frac{\partial^2F(V_3)}{\partial(V_2-V_3)\partial(V_1-V_3)}+\bar{h}_1'(1)\frac{\partial F(V_3)}{\partial(V_3-V_2)} \\
&\quad +\bar{h}_1'(1)\left[\frac{\partial^2F(V_3)}{\partial(V_1-V_3)\partial(V_3-V_2)}-\frac{\partial F(V_3)}{\partial(V_3-V_2)}\right]\Bigg] \\
&= \bar{h}_0(1-S)\left[\frac{\partial^2F(V_3)}{\partial(V_2-V_3)\partial(V_1-V_3)}+\frac{\partial^2F(V_3)}{\partial(V_1-V_3)\partial(V_3-V_2)}\right].
\end{aligned}
\tag{2.7}
$$

Hence

$$
\begin{aligned}
& P_3^2P_2^1(I-P_1)F \\
&= \bar{h}_0\left(\frac{b_2}{1-b_3}\right)(1-b_3)\bar{h}_0(1-b_3)\left[\frac{\partial^2F(V_3)}{\partial(V_2-V_3)\partial(V_1-V_3)}+\frac{\partial^2F(V_3)}{\partial(V_1-V_3)\partial(V_3-V_2)}\right].
\end{aligned}
$$

The triple Boolean sum $P_3 \oplus P_2 \oplus P_1$ is given by substitution of equations (2.4), (2.6), and (2.7) into (2.5).

(ii) Discretization. The triple Boolean sum interpolant $P_3 \oplus P_2 \oplus P_1$

requires data all along the edge of a triangle. These data can be manufactured from vertex and other point data using univariate interpolation. This procedure is called discretizing the interpolant. It reduces a transfinite interpolant to one with a finite number of parameters. The composition of an interpolant with this discretizing process produces a greatly simplified interpolant. Let D be the final discretized interpolant. The process of inserting the discretizing univariate interpolants into the transfinite formula can be formalized by $D = (P_3 \oplus P_2 \oplus P_1)D$. This circular statement has great utility for the following trick. Let D have cubic Hermite edge values and quadratic derivative in the direction normal to each edge along the edge. The value of the normal directional derivative at the midpoints of each of the edges and the value and gradient at the vertices (twelve parameters) supply the data to define uniquely these interpolants. Let D (the discretization of $(P_3 \oplus P_2 \oplus P_1)$) interpolate these quantities. If C is a cubic polynomial, then $DC = (P_3 \oplus P_2 \oplus P_1)DC = (P_3 \oplus P_2 \oplus P_1)C = C$, since C already has cubic edge values and quadratic normal derivatives and the Boolean sum has the cubic precision of P_1. Let

$$C^9F = \sum_{i=1}^{3} b_i^2\left\{(3-2b_i)F(V_i) + \sum_{\substack{j=1\\ j\neq i}}^{3} b_j \frac{\partial F(V_i)}{\partial(V_j - V_i)}\right\}, \tag{2.8}$$

and

$$DF = DF + C^9F - C^9F = C^9F + D(F - C^9F). \tag{2.9}$$

Further, $D(F - C^9F)$ has zero edge value, one-parameter quadratic normal derivatives, i.e.,

$$\frac{\partial}{\partial n_i} D(F - C^9F)\bigg|_{(1-S)V_j+SV_k} = \frac{\partial}{\partial n_i}(F - C^9F)\bigg|_{(V_j+V_k)/2} \cdot 4S(1 - S),$$

and zero gradient at the vertices. The algebra of inserting these values into the formula of $P_3 \oplus P_2 \oplus P_1$ is now feasible and gives

$$\begin{aligned} DF = C^9F + 4b_1b_2b_3\Bigg(&N_1b_2b_3\left\{\frac{1}{1-b_2} + \frac{1}{1-b_3}\right\} \\ &+ N_2b_1b_3\left\{\frac{1}{1-b_1}+\frac{1}{1-b_3}\right\}+N_3b_1b_2\left\{\frac{1}{1-b_1}+\frac{1}{1-b_2}\right\}\Bigg) \end{aligned} \tag{2.10}$$

where

$$N_i = \frac{1}{\dfrac{\partial b_i}{\partial n_i}} \frac{\partial}{\partial n_i}(F - C^9F)\bigg|_{(V_j+V_k)/2}.$$

For implementation, we note that

$$\frac{1}{\frac{\partial b_1}{\partial n_1}} \frac{\partial C^9 F}{\partial n_1}\bigg|_{(V_2+V_3)/2} = \frac{3}{2}((\alpha - 1)F(V_2) - \alpha F(V_3))$$

$$+\frac{1}{4}\left(\frac{\partial F(V_2)}{\partial(V_3 - V_2)}(\alpha - 2) - \frac{\partial F(V_3)}{\partial(V_2 - V_3)}(\alpha + 1) + \frac{\partial F(V_3)}{\partial(V_1 - V_3)} + \frac{\partial F(V_2)}{\partial(V_2 - V_1)}\right) \tag{2.11}$$

where $\alpha = (V_1 - V_2) \cdot (V_3 - V_2)/|V_3 - V_2|^2$. Substitution of equations (2.8), (2.10), and (2.11) into (2.9) yields the discretization DF.

(2) Four-dimensional BBG schemes.

(i) Transfinite. Let $n = 3$ and choose the edge contained in $\{b_3 = 0\}$ and $\{b_4 = 0\}$ which is $\overline{V_1V_2}$. Let $V = b_1V_1 + b_2V_2 + b_3V_3 + b_4V_4$, $A = (b_1 + b_2)V_1 + b_3V_3 + b_4V_4$, $B = (b_1 + b_2)V_2 + b_3V_3 + b_4V_4$. Notice that $b_2(A) = 0, b_1(B) = 0$ and that $\overline{AB}$ is in the same contour of b_3 and b_4 as V is. Since

$$V = \frac{b_1}{b_1 + b_2} A + \frac{b_2}{b_1 + b_2} B,$$

t is $b_2/(b_1 + b_2)$ and $B - A = (b_1 + b_2)(V_2 - V_1)$. The BBG projector on a tetrahedron is

$$\begin{aligned} P_{12}(F; V) &= h_0\left(\frac{b_2}{b_1+b_2}\right)F((b_1 + b_2)V_1 + b_3V_3 + b_4V_4) \\ &+ h_1\left(\frac{b_2}{b_1+b_2}\right)F((b_1 + b_2)V_2 + b_3V_3 + b_4V_4) \\ &+ \bar{h}_0\left(\frac{b_2}{b_1+b_2}\right)(b_1 + b_2)\frac{\partial F((b_1+b_2)V_1+b_3V_3+b_4V_4)}{\partial(V_2 - V_1)} \\ &+ \bar{h}_1\left(\frac{b_2}{b_1+b_2}\right)(b_1+b_2)\frac{\partial F((b_1+b_2)V_2+b_3V_3+b_4V_4)}{\partial(V_2 - V_1)}. \end{aligned} \tag{2.12}$$

Let us introduce some notation. In the context of a fixed simplex for a function F define a function $\hat{F}$: $\mathbf{R}^{n+1} \to \mathbf{R}$ by

$$\hat{F}(b_1, b_2, \ldots, b_{n+1}) = F(b_1V_1 + b_2V_2 + \cdots + b_{n+1}V_{n+1}). \tag{2.13}$$

Since $\partial b_i/\partial(V_k - V_j) = \delta_{ik} - \delta_{ij}$, an application of the Chain Rule gives $\partial F/\partial(V_k - V_j) = \partial\hat{F}/\partial b_k - \partial\hat{F}/\partial b_j$. This equation is useful in the composition of projectors. Also define $|_V\ F = F(V)$ and $\uparrow_{(b_1, b_2, b_3, b_4)} F = \hat{F}(b_1, b_2, b_3, b_4)$. The BBG Projector P_{12} can now be rewritten in matrix form as

$$\begin{aligned} P_{12} = [h_0, h_1, \bar{h}_0, \bar{h}_1]\Big\uparrow_{b_2/(b_1+b_2)} \Big[&\Big\uparrow_{b_1+b_2, 0, b_3, b_4}, \Big\uparrow_{0, b_1+b_2, b_3, b_4}, \\ &(b_1+b_2)\Big\uparrow_{b_1+b_2, 0, b_3, b_4} \frac{\partial}{\partial(V_2 - V_1)}, (b_1+b_2)\Big\uparrow_{0, b_1+b_2, b_3, b_4} \frac{\partial}{\partial(V_2 - V_1)}\Big]^t F. \end{aligned} \tag{2.14}$$

Since $\partial(b_3 + b_4)/\partial(V_2 - V_1) = 0$, $P_{12} \oplus P_{34} = I - (I - P_{12}) \cdot (I - P_{34})$ can be written in the following matrix form:

$$(2.15) \qquad I - [-1, h_0, h_1, \bar{h}_0, \bar{h}_1]\Big|_{b_2/(b_1+b_2)} \mathbf{P}_{5\times 5}[-1, h_0, h_1, \bar{h}_0, \bar{h}_1]^t\Big|_{b_4/(b_3+b_4)}$$

where $\mathbf{P}_{5\times5}$ is the 5×5 matrix from the product of

$$\Big[I, \Big|^{\uparrow}_{b_1+b_2, 0, b_3, b_4}, \Big|^{\uparrow}_{0, b_1+b_2, b_3, b_4}, (b_1 + b_2)\Big|^{\uparrow}_{b_1+b_2, 0, b_3, b_4} \frac{\partial}{\partial(V_2 - V_1)}, \\ (b_1 + b_2)\Big|^{\uparrow}_{0, b_1+b_2, b_3, b_4} \frac{\partial}{\partial(V_2 - V_1)}\Big]^t$$

and

$$\Big[I, \Big|^{\uparrow}_{b_1, b_2, b_3+b_4, 0}, \Big|^{\uparrow}_{b_1, b_2, 0, b_3+b_4}, (b_3 + b_4)\Big|^{\uparrow}_{b_1, b_2, b_3+b_4, 0} \frac{\partial}{\partial(V_4 - V_3)}, \\ (b_3 + b_4)\Big|^{\uparrow}_{b_1, b_2, 0, b_3+b_4} \frac{\partial}{\partial(V_4 - V_3)}\Big]F.$$

The Boolean sum $P_{12} \oplus P_{34}$ has a twist compatibility condition all along $\overline{V_1V_3}$, $\overline{V_1V_4}$, $\overline{V_2V_3}$ and $\overline{V_2V_4}$, namely,

$$\frac{\partial}{\partial(V_4 - V_3)}(P_{12} \oplus P_{34})F = \frac{\partial}{\partial(V_4 - V_3)}F$$

on $\{b_3 = 0, b_4 = 0\}$ if and only if

$$\frac{\partial}{\partial(V_2 - V_1)} \frac{\partial}{\partial(V_4 - V_3)} F = \frac{\partial}{\partial(V_4 - V_3)} \frac{\partial}{\partial(V_2 - V_1)} F$$

on the above four edges. The compatibility condition is removed in the following manner similar to **[10]**. In the matrix $P_{5\times5}$ identify the lower right 2×2 submatrix and replace it with the following convex combination matrix S where

$$S_{11} = \frac{(b_1 + b_2)(b_3 + b_4)}{b_2 + b_4} \\ \cdot \Big[b_4 \Big|^{\uparrow}_{b_1+b_2, 0, b_3+b_4, 0} \frac{\partial}{\partial(V_2 - V_1)} \frac{\partial}{\partial(V_4 - V_3)} \\ + b_2 \Big|^{\uparrow}_{b_1+b_2, 0, b_3+b_4, 0} \frac{\partial}{\partial(V_4 - V_3)} \frac{\partial}{\partial(V_2 - V_1)}\Big]$$

$$(2.16) \qquad S_{21} = \frac{(b_1 + b_2)(b_3 + b_4)}{b_1 + b_4} \\ \cdot \Big[b_4 \Big|^{\uparrow}_{0, b_1+b_2, b_3+b_4, 0} \frac{\partial}{\partial(V_2 - V_1)} \frac{\partial}{\partial(V_4 - V_3)} \\ + b_1 \Big|^{\uparrow}_{0, b_1+b_2, b_3+b_4, 0} \frac{\partial}{\partial(V_4 - V_3)} \frac{\partial}{\partial(V_2 - V_1)}\Big]$$

$$S_{12} = \frac{(b_1 + b_2)(b_3 + b_4)}{b_2 + b_3} \cdot \left[b_3 \Big\uparrow_{b_1+b_2,0,0,b_3+b_4} \frac{\partial}{\partial(V_2 - V_1)} \frac{\partial}{\partial(V_4 - V_3)} + b_2 \Big\uparrow_{b_1+b_2,0,0,b_3+b_4} \frac{\partial}{\partial(V_4 - V_3)} \frac{\partial}{\partial(V_2 - V_1)} \right]$$

and

$$S_{22} = \frac{(b_1 + b_2)(b_3 + b_4)}{b_1 + b_3} \cdot \left[b_3 \Big\uparrow_{0,b_1+b_2,0,b_3+b_4} \frac{\partial}{\partial(V_2 - V_1)} \frac{\partial}{\partial(V_4 - V_3)} + b_1 \Big\uparrow_{0,b_1+b_2,0,b_3+b_4} \frac{\partial}{\partial(V_4 - V_3)} \frac{\partial}{\partial(V_2 - V_1)} \right].$$

The resulting C^1 transfinite interpolant on a tetrahedron, called $(P_{12} \oplus P_{34})^+$, is free of compatibility conditions.

(ii) Discretization. Discretizations of transfinite interpolants fill the place in piecewise triangular interpolation that the cubic Hermite interpolant holds in curve fitting. On one tetrahedron in a triangulation a discretization matches data given at vertices and, more importantly, maintains appropriate continuity with the interpolant on the next tetrahedron. This region-to-region continuity molds the discretization. C^1 continuity across a common vertex forces joint interpolation to position and three directional derivatives. Given these data which define a unique cubic Hermite value along an edge, it seems natural to require all interpolants on this edge to match the given cubic value. The data also define a unique linear interpolant to all directional derivatives perpendicular to the edge. Experience with triangular schemes has shown that supplying a midpoint value and using a quadratic interpolant is more general and actually simplifies the algebra. The linear interpolant can be obtained by supplying the average of the endpoint values at the midpoint. This is an instance of "reduction of parameters" in finite element analysis. Many triangular interpolants have the above vertex and edge behavior and so are appropriate to use as the value of restriction of the discretized tetrahedral interpolant to one of its faces. The simplest is perhaps the Madison triangle. The across-face derivative is to have quadratic edge restriction and so the six-degrees-of-freedom bivariate quadratic interpolant is appropriate. The discussion above is to motivate the definition of the 28 parameter tetrahedral interpolant D, which is the discretization of $(P_{12} \oplus P_{34})^+$, i.e.,

$$D = (P_{12} \oplus P_{34})^+ D. \tag{2.17}$$

The interpolant D restricted to a face is the Madison triangular interpolant and has a quadratic cross-face derivative. The 28 interpolated parameters are position and three directional derivatives at each of four vertices and two cross-edge directional derivatives at the midpoints of the six edges. This discretization has inherent incompatibilities in the twists along the edges.

Traditionally, discretization of a transfinite interpolant is accomplished by substituting the discretizing interpolants into the interpolation formula in place of transfinite evaluations. If one is programming this for computer evaluation, this is straightforward. One is tempted to substitute and simplify mathematically but this is a tedious method. Alternatively, consider the following nifty trick.

First observe that if C is cubic, it already has discretized value and derivatives, i.e.,

$$DC = (P_{12} \oplus P_{34})^+ DC = (P_{12} \oplus P_{34})^+ C \tag{2.18}$$

and since the transfinite interpolant has cubic precision, $DC = C$.

We construct a 20 (of 28) parameter cubic interpolant C^{20} which matches function and gradient at the vertices and matches the cross-edge derivatives at the midpoints of two skew edges. We now pull the trick.

$$DF = DF + C^{20}F - C^{20}F = C^{20}F + D(F - C^{20}F) \qquad (= (D \oplus C^{20})F). \tag{2.19}$$

This has separated the interpolation problem into a polynomial and an eight-degrees-of-freedom correction interpolant (a la Coons [7]). To shorten the formulas describing the 28 parameter interpolant, let

$$F_i = F(V_i),\ F_{ij} = \frac{\partial F(V_i)}{\partial (V_j - V_i)},$$

$$V_{ijk} = (V_k - V_i) - \frac{((V_k - V_i) \cdot (V_j - V_i))(V_j - V_i)}{\|V_j - V_i\|^2},$$

$$D_{ijk} = \left.\frac{\partial}{\partial V_{ijk}}\right|_{(V_i+V_j)/2} \text{ and } N_{ijk} = \left.\frac{\partial}{\partial n_k}\right|_{(V_i+V_j)/2},$$

where n_k is the unit inward normal to the k-th face. Define the following cubic interpolants:

$$C^{16}F = \sum_{i=1}^{4} \Big[(3 - 2b_i)F_i + \sum_{\substack{j=1\\ j\neq i}}^{4} b_j F_{ij}\Big] b_i^2 \tag{2.20}$$

$$\begin{aligned} C^{20}F = C^{16}F &+ 4b_1b_2[b_3D_{123}(F - C^{16}F) + b_4D_{124}(F - C^{16}F)] \\ &+ 4b_3b_4[b_1D_{341}(F - C^{16}F) + b_2D_{342}(F - C^{16}F)]. \end{aligned} \tag{2.21}$$

The term $D_{ijk}C^{16}$ is a fixed linear combination of the 16 parameters of $C^{16}F$. Considering the b_i as functions, we have

$$D_{123}(b_1, b_2, b_3, b_4) = (\alpha - 1, -\alpha, 1, 0) \tag{2.22}$$

where

$$\alpha = (V_3 - V_1) \cdot (V_2 - V_1)/\|V_2 - V_1\|^2. \tag{2.23}$$

Then

$$D_{123}C^{16} = 3(F_1(\alpha - 1) - F_2\alpha)/2 + (F_{12}(\alpha - 2) - F_{21}(\alpha + 1) + F_{13} + F_{23})/4,$$

Dually

$$\begin{aligned} D_{124}C^{16} &= 3(F_1(\beta - 1) - F_2\beta)/2 \\ &\quad + (F_{12}(\beta - 2) - F_{21}(\beta + 1) + F_{14} + F_{24})/4, \\ D_{341}C^{16} &= 3(F_3(\gamma - 1) - F_4\gamma)/2 \\ &\quad + (F_{34}(\gamma - 2) - F_{43}(\gamma + 1) + F_{31} + F_{41})/4, \\ D_{342}C^{16} &= 3/(F_3(\delta - 1) - F_4\delta)/2 \\ &\quad + (F_{34}(\delta - 2) - F_{43}(\delta + 1) + F_{32} + F_{42})/4, \\ D_{134}C^{20}F &= D_{134}C^{16} - D_{341}(F - C^{16}F) \end{aligned} \tag{2.24}$$

where

$$\alpha = \frac{(V_3 - V_1) \cdot (V_2 - V_1)}{\|V_2 - V_1\|^2}, \qquad \beta = \frac{(V_4 - V_1) \cdot (V_2 - V_1)}{\|V_2 - V_1\|^2}$$

$$\gamma = \frac{(V_1 - V_3) \cdot (V_4 - V_3)}{\|V_4 - V_3\|^2} \text{ and } \delta = \frac{(V_2 - V_3) \cdot (V_4 - V_3)}{\|V_4 - V_3\|^2}.$$

The 28 parameters are the F_i, F_{ij} and $D_{ijk}F$. For (i, j, k, l) a permutation of (1, 2, 3, 4),

$$V_{ijk} = n_k(n_k \cdot V_{ijk}) + V_{ijl}\frac{V_{ijk} \cdot V_{ijl}}{|V_{ijl}|^2}. \tag{2.25}$$

Let

$$\tilde{n}_k = V_{ijk} - V_{ijl}\frac{(V_{ijk} \cdot V_{ijl})}{|V_{ijl}|^2} \tag{2.26}$$

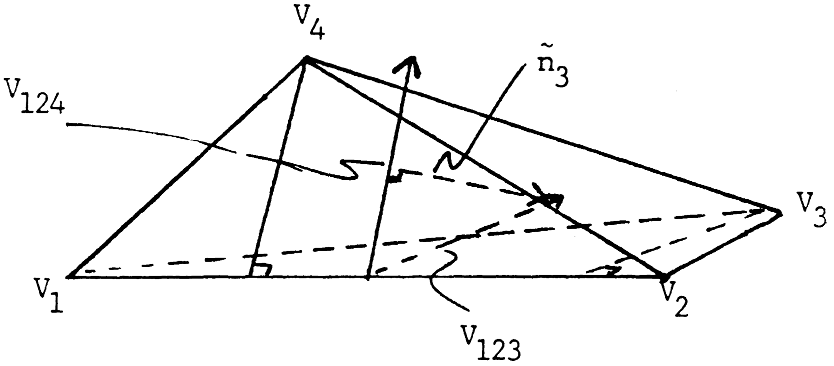
and

$$n_k = \tilde{n}_k/|\tilde{n}_k|. \tag{2.27}$$

Therefore we find that

$$N_{ijk} = \frac{1}{|\tilde{n}_k|}\left[D_{ijk} - D_{ijl}\frac{V_{ijk} \cdot V_{ijl}}{|V_{ijl}|^2}\right]. \tag{2.28}$$

For piecewise interpolation the various directional derivatives at the midpoints of the edges must be derived in this manner from two independent directional derivatives in directions perpendicular to the edge. Reduction of parameters to a 16-degrees-of-freedom interpolant eliminates this complexity.



Geometry for BBG Tetrahedral Interpolant's Discretization

The discretizing interpolants on face one are the following:

$$D(I-C^{20})\Big|_{0,r,s,t} = \left(\frac{4r^2rt}{st+rt+rs}\right)(sD_{231}(I-C^{20})+tD_{241}(I-C^{20})), \tag{2.29}$$

and

$$\frac{\partial D(I-C^{20})}{\partial n_1}\Big|_{0,r,s,t} = 4r[sN_{231}(I-C^{20}) + tN_{241}(I-C^{20})] \tag{2.30}$$

with analogous discretizations on the other three faces. The quantities needed in the transfinite interpolant are derived from these interpolants. First we resolve $V_2 - V_1$ into its components perpendicular to face one and in face one, respectively, as follows:

$$(V_2-V_1)=n_1(n_1\cdot(V_2-V_1))+[(V_2-V_1)-n_1(n_1\cdot(V_2-V_1))]. \tag{2.31}$$

Then the derivative in the direction of $V_2 - V_1$ consists of two terms. Since the second term involves a derivative in face one, evaluations on that face commute with the derivative which leads to

$$\begin{aligned}\frac{\partial D(I-C^{20})}{\partial(V_2-V_1)}\Big|_{0,r,s,t} &= n_1\cdot(V_2-V_1)\frac{\partial D(I-C^{20})}{\partial n_1}\Big|_{0,r,s,t}\\ &+\frac{\partial}{\partial r}\left[D(I-C^{20})\Big|_{0,r,s,t}\right]-n_1\cdot(V_2-V_1)\frac{\partial}{\partial n_1}\left[D(I-C^{20})\Big|_{0,r,s,t}\right]\end{aligned} \tag{2.32}$$

Working out each part yields

$$\frac{\partial}{\partial r}\left[D(I - C^{20})\uparrow\Big|_{0,r,s,t}\right] \tag{2.33}$$
$$= \frac{4rst(2st + rt + rs)}{(st + rt + rs)^2}(sD_{231}(I - C^{20}) + tD_{241}(I - C^{20}))$$

and

$$\begin{aligned}\frac{\partial}{\partial n_1}\left[D(I-C^{20})\uparrow\Big|_{0,r,s,t}\right] &= D_{231}(I-C^{20})\left[\frac{\partial b_2}{\partial n_1}\cdot 4rs^2t(rs+rt+2st)\right.\\ &\quad \left.+\frac{\partial b_3}{\partial n_1}4r^2st(rs + 2rt + st) + \frac{\partial b_4}{\partial n_1}4r^3s^3\right]\Big/(st + rt + rs)^2\\ &\quad + D_{241}(I - C^{20})\left[\frac{\partial b_2}{\partial n_1}4rst^2(rs + rt + 2st) + \frac{\partial b_3}{\partial n_1}4r^3t^3\right.\\ &\quad \left.+\frac{\partial b_4}{\partial n_1}\cdot 4r^2st(2rs + rt + st)\right]\Big/(st + rt + rs)^2.\end{aligned} \tag{2.34}$$

A typical derivative is $\partial b_2/\partial n_1 = \hat{n}_2 \cdot n_1/6V$ where $\hat{n}_2 = (V_1 - V_4) \times (V_3 - V_4)$ and

$$6V = \begin{vmatrix} V_1 & V_2 & V_3 & V_4 \\ 1 & 1 & 1 & 1 \end{vmatrix} = \pm 6 \cdot \text{Volume of simplex.}$$

Note that $\partial b_3/\partial n_1$ and $\partial b_4/\partial n_1$ are dual. Also needed in the interpolation formula are the edge values for derivatives, for example,

$$\frac{\partial(I - C^{20})\uparrow}{\partial(V_2 - V_1)}\Big|_{0,r,0,t} = n_1\cdot(V_2 - V_1)4rt(N_{241}(I - C^{20}) - \frac{\partial b_3}{\partial n_1}D_{241}(I - C^{20})). \tag{2.35}$$

Also needed are the mixed partials along these edges:

$$\begin{aligned}&\frac{\partial}{\partial(V_4 - V_3)}\frac{\partial(I - C^{20})\uparrow}{\partial(V_2 - V_1)}\Big|_{0,r,0,t}\\ &= n_1 \cdot (V_2 - V_3)4r[N_{241}(I - C^{20}) - N_{231}(I - C^{20})]\\ &\qquad + 4tD_{241}(I - C^{20}) - n_1 \cdot (V_2 - V_3)\\ &\qquad \cdot\left[D_{231}(I - C^{20}) - 8\frac{\partial b_3}{\partial n_1}r + D_{241}(I - C^{20})\right.\\ &\qquad \left. 4\left\{-t\frac{\partial b_2}{\partial n_1} + (3r + 2t)\frac{\partial b_3}{\partial n_1} - r\frac{\partial b_4}{\partial n_1}\right\}\right].\end{aligned} \tag{2.36}$$

The final result D^{28} is obtained via the substitution of the interpolated quantities from (2.32–2.36) into (2.19) using (2.17), (2.16) and (2.15). This is best done in an algorithmic setting which automates the symmetry of (2.32–2.36).

B. Radial Nielson interpolants.

(1) n-dimensional Nielson schemes. Nielson's radial projector [15] is cubic Hermite interpolation where one end of the segment is a vertex of the simplex and the other is the (radial) projection of the point of evaluation onto the opposite face. Consider the point

$$V = \sum_{i=1}^{n+1} V_i b_i = b_1 V_1 + (1-b_1) \sum_{i=2}^{n+1} \frac{b_i}{1-b_1} V_i \qquad (=(1-t)A+tB). \tag{2.37}$$

The radial projector $Q_1 F$ is given by

$$\begin{aligned} Q_1 F \equiv{}& h_0(b_1) F\left(\sum_{i=2}^{n+1} \frac{b_i}{1-b_1} V_i\right) + h_1(b_1) F(V_1) \\ &+ \bar{h}_0(b_1) \frac{\partial F\left(\sum_{i=2}^{n} \frac{b_i}{1-b_1} V_i\right)}{\partial\left(V_1 - \sum_{i=2}^{n} \frac{b_i}{1-b_1} V_i\right)} + \bar{h}_1(b_1) \frac{\partial F(V_1)}{\partial\left(V_1 - \sum_{i=2}^{n} \frac{b_i}{1-b_1} V_i\right)}. \end{aligned} \tag{2.38}$$

The projector $Q_1 F$ interpolates F and its radial derivatives at V_1 and all across the opposite face. Let $Q_1 = S_1 + C_1$ where C_1 is the sum of all the terms evaluated at V_1 and is cubic (if $\nabla F(V_1)$ exists) since $h_1(b_1)F(V_1)$ certainly is cubic and

$$\begin{aligned} \bar{h}_1(b_1) \frac{\partial F(V_1)}{\partial\left(V_1 - \sum_{i=2}^{n} \frac{b_i}{1-b_1} V_i\right)} &= (b_1 - 1) b_1^2 \sum_{i=2}^{n} \frac{b_i (V_1 - V_i)}{(1-b_1)} \cdot \nabla F(V_i) \\ &= -b_i^2 \sum_{i=2}^{n} b_i \frac{\partial F(V_i)}{\partial (V_1 - V_i)} \end{aligned}$$

is cubic. Note that Q_1 has cubic precision since $Q_1 F(V) - F(V)$ restricted to the radial segment is just $H_3(F; V) - F$.

As in the BBG discussion the plan is to take the union of the interpolation properties of all Q_i via some combination. The following theorem describes a simple way of doing this. Let D^r be an arbitrary Gateaux derivative of order r. Let $\{P_i F\}_{i=1}^{n+1}$ be a set of interpolants such that

$$D^r P_i F\Big|_{\{b_i=0\}} = D^r F \qquad \text{for all } r,\ 0 \leqq r \leqq q, \text{ and}$$

$$D^r P_i F\Big|_{\text{all faces}} = D^r F \qquad \text{for all } r,\ 0 \leqq r \leqq p,\ 0 \leqq p \leqq q.$$

Let $\{\lambda_i\}$ be a collection of functions such that $\sum_{i=1}^{n+1} \lambda_i = 1$ and $D^r \lambda_i|_{\{b_j=0\}} \equiv 0$ for all $j \neq i$, $0 \leqq r < q - p$. The conclusion is that the combination $KF = \sum_{i=1}^{n+1} \lambda_i P_i F_i$ has the union of the interpolation properties of the P_i, i.e., $D^r KF = D^r F$ on all faces, $0 \leqq r \leqq q$.

We define the *convex combination* Radial Interpolant

$$QF = \sum_{i=1}^{n+1} \lambda_i Q_i F \text{ where } \lambda_i = \left(\frac{1}{b_i}\right)^2 \Big/ \sum_{j=1}^{n+1} \left(\frac{1}{b_j}\right)^2. \tag{2.39}$$

This rational function only blows up on the boundary of the simplex and $QF = F$ there by interpolation. (Multiplying λ_i by

$$1 = \prod_{k=1}^{n+1} b_k^2 \Big/ \prod_{k=1}^{n+1} b_k^2$$

provides weights which blow up only at the vertices but this form is computationally expensive and noisy.)

(2) Three-dimensional radial schemes. Let us explore the triple Boolean sum of radial projectors on the triangle. Some observations help in this endeavor. As discussed above, the C_i are cubic. In the composition $S_j S_k$ the functionals which emerge are of order two or less; so if T_i is a quadratic Taylor interpolant at V_i, then $S_j S_k F = S_j S_k T_i F$. Substituting $S_j = Q_j - C_j$ gives $S_j S_k F = Q_j Q_k T_i F - Q_j C_k T_i F - C_j Q_k T_i F + C_j C_k T_i F$. The cubic precision of Q and Q_k shows that $S_j S_k F$ is a cubic polynomial. The two terms of C_i each contain a factor of b_i^2. This second order zero on the opposite edge gives $C_i C_j = \delta_{ij} C_i$. This fact and the cubic precision of Q_i gives $S_i C_j = C_j, j \neq i$. All these observations induce cancellations in the expansion of $Q_1 \oplus Q_2 \oplus Q_3$ and ultimately

$$\begin{aligned} Q_1 \oplus Q_2 \oplus Q_3 &= S_1 \oplus S_2 \oplus S_3 \\ &= \sum_{i=1}^{3} (S_i - C_i) - S_1 S_2 (I - C_3) - S_1 S_3 (I - C_2). \end{aligned} \tag{2.40}$$

Observe that $S_1 S_2 (I - C_3)$ and $S_1 S_3 (I - C_2)$ are both cubic polynomials which are zero and have zero gradient at all three vertices. The only cubic with this property is $b_1 b_2 b_3$; so

$$\bigoplus_{i=1}^{3} Q_i = \sum_{i=1}^{3} (S_i - C_i) + b_1 b_2 b_3 k_1. \tag{2.41}$$

The factor k_1 has yet to be determined so recall that $\oplus_{i=1}^{3} Q_i$ has the interpolation properties of Q_1. In particular the linear functional

$$L = \frac{1}{\left(\dfrac{\partial b_1}{\partial n_1}\right)} \frac{\partial}{\partial n_1}\bigg|_{(V_2+V_3)/2}.$$

is interpolated by Q_1. Therefore,

$$k_1 = 4\left(LF - L \sum_{i=1}^{3} (S_i - C_i)\right) = 4(LC_2 + LC_3 - LS_2 - LS_3).$$

The terms LC_2 and LC_3 are in equation (2.11) of the BBG discussion; so we need only calculate

$$LS_2 = \frac{3(\alpha-1)}{2}F(V_3) + \frac{(\alpha-1)}{4}\frac{\partial F(V_3)}{\partial(V_2-V_3)} + \frac{1}{2}\frac{\partial F(V_3)}{\partial(V_1-V_3)} + \frac{1}{8}\frac{\partial^2 F(V_3)}{\partial(V_1-V_3)\partial(V_2-V_3)}$$

where $\alpha = (V_1 - V_2)(V_3 - V_2)/|V_3 - V_2|^2$ and

$$LS_3 = -\frac{3\alpha}{2}F(V_2) - \frac{\alpha}{4}\frac{\partial F(V_2)}{\partial(V_2-V_3)} + \frac{1}{2}\frac{\partial F(V_2)}{\partial(V_1-V_2)} + \frac{1}{8}\frac{\partial^2 F(V_2)}{\partial(V_1-V_2)\partial(V_2-V_3)}.$$

The final formula is symmetric in the last two indices, i.e., $Q_1 \oplus Q_2 \oplus Q_3 = Q_1 \oplus Q_3 \oplus Q_2$. The formulae for the Boolean sums with Q_2 and Q_3 first are identical to that for Q_1 first except for the index on k. This allows us to tie down the interpolation properties of the triple Boolean sum. By the interpolation part of the Barnhill-Gregory Theorem,

$$(F - Q_1 \oplus Q_2 \oplus Q_3)|_{b_2=0} = (Q_2 \oplus Q_1 \oplus Q_3 - Q_1 \oplus Q_2 \oplus Q_3|_{b_2=0}.$$

As noted above, the latter equals $b_1b_2b_3(k_2 - k_1)$ which, evaluated along $b_2 = 0$, is zero. For a cross-boundary derivative D on side two, $Q_1 \oplus Q_2 \oplus Q_3$ fails to interpolate since $D(F - Q_1 \oplus Q_2 \oplus Q_3) = (Db_2)b_1b_3(k_2 - k_1) \neq 0$.

The convex combination theorem is now used on the Boolean sum radial projectors to create the following C^1 interpolant:

$$Q = \sum_{i=1}^{3}(S_i - C_i) + b_1b_2b_3\frac{(k_1/b_1 + k_2/b_2 + k_3/b_3)}{1/b_1 + 1/b_2 + 1/b_3}. \tag{2.42}$$

The convex combination theorem can also be used to create a cubic polynomial interpolant as follows. Let

$$C^9F = \sum_{i=1}^{3} C_iF = \sum_{i=1}^{3} b_i^2\left[(3-2b_i)F(V_i) + \sum_{\substack{j=1\\ j\neq i}}^{3} b_j\frac{\partial F(V_i)}{\partial(V_j - V_i)}\right]. \tag{2.43}$$

and

$$C_1^{10}F = C^9F + 4b_1b_2b_3\left(\frac{\partial b_1}{\partial n_1}\right)^{-1}\frac{\partial}{\partial n_1}(F - C^9F)\bigg|_{(V_2+V_3)/2}. \tag{2.44}$$

Then C_1^{10} has a cubic Hermite edge value on the boundary of the triangle and a quadratic cross-boundary derivative. The "Madison triangle" is

$$
\begin{aligned}
C^{12} &= \left(\sum_{i=1}^{3} C_i^{10}/b_i\right) \Big/ \left(\frac{1}{b_1} + \frac{1}{b_2} + \frac{1}{b_3}\right) \\
&= C^9 + 4b_1b_2b_3 \left(\frac{\sum_{i=1}^{3} N_i/b_i}{\sum_{i=1}^{3} 1/b_i} \right)
\end{aligned} \tag{2.45}
$$

where

$$
N_1 = \left(\frac{\partial b_1}{\partial n_1}\right)^{-1} \frac{\partial}{\partial n_1}(F - C^9F)\Big|_{(V_2+V_3)/2},
$$

etc. Then C^{12} used as a piecewise scheme produces a C^1 surface. A nine-parameter version is available through reduction of parameters, i.e., by defining N_1 as

$$
N_1 = \left(\frac{\partial b_1}{\partial n_1}\right)^{-1} \left[\frac{1}{2}\left(\frac{\partial F(V_2)}{\partial n_1} + \frac{\partial F(V_3)}{\partial n_1} \right) - \frac{\partial C^9 F}{\partial n_1}\Big|_{(V_2+V_3)/2} \right], \tag{2.46}
$$

etc. Thus C^{12} is the discretization of Q.

Bibliography

1. R. E. Barnhill, *Representation and approximation of surfaces*, Mathematical Software III, J. R. Rice, editor, Academic Press, NY, 1977, 69–120.

2. ———, *Computer aided surface representation and design*, Surfaces in Computer Aided Geometric Design, R. E. Barnhill and W. Boehm, editors, North-Holland, 1983.

3. ———, G. Birkhoff and W. J. Gordon, *Smooth interpolation in triangles*, Journal of Approximation Theory **8** (1973), 114–128.

4. R. E. Barnhill, R. P. Dube and F. F. Little, *Properties of Shepard's surfaces*, Rocky Mountain Journal of Mathematics **13** (1983), 365–382.

5. R. E. Barnhill and J. A. Gregory, *Polynomial interpolation to boundary data on triangles*, Mathematics of Computation **29** (1975), no. 131, 726–735.

6. R. E. Barnhill and S. E. Stead, *Multistage trivariate Surfaces*, this Volume.

7. S. A. Coons, *Surfaces for Computer Aided Design*, Design Division, Mech. Engin. Dept., M.I.T., revised, 1967. (Available as AD 663 504 from CFSTI, Sills Building, 5285 Port Royal Road, Springfield, VA 22151).

8. D. J. F. Ewing, A. J. Fawkes and J. R. Griffiths, *Rules governing the numbers of nodes and elements in a finite element mesh*, International Journal for Numerical Methods in Engineering **2** (1970), 597–601.

9. P. J. Green and R. Sibson, *Computing Dirichlet tessellations in the plane*, The Computer Journal **21** (1978), 168–173.

10. J. A. Gregory, *Smooth interpolation without twist constraints*, Computer Aided Geometric Design, R. E. Barnhill and R. F. Riesenfeld, editors, Academic Press, NY, 1975.

11. ———, *Error bounds for linear interpolation on triangles*, The Mathematics of Finite Elements and Application II, J. R. Whiteman, editor, Academic Press, London, 1975, 163–170.

12. C. L. Lawson, *Software for C^1 surface interpolation*, Mathematical Software II, J. R. Rice, editor, Academic Press, NY, 1977, 161–194.

13. ———, C^1 surface interpolations for scattered data on a sphere, this volume.

14. F. F. Little, *Convex combination surfaces*, Surfaces in Computer Aided Geometric Design, R. E. Barnhill and W. Boehm, editors, North-Holland, 1983.

15. G. M. Nielson, *The side-vertex method for interpolation in triangles*, Journal of Approximation Theory **25** (1979), 318–336.

16. C. S. Petersen, *Contours of three and four-dimensional surfaces*, Masters Thesis, Department of Mathematics, University of Utah, 1983.

17. R. Sibson, *Locally equiangular triangulations*, The Computer Journal **21** (1978), 243–245.

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF UTAH, SALT LAKE CITY, UT 84112

LOCKHEED-CALIFORNIA COMPANY, DEPT. 75–25, BLDG. 90, PLANT A1, BOX 551, BURBANK, CA 91520