Scattered Data Interpolation in Three or More Variables

by

Peter Alfeld

Abstract

This is a survey of techniques for the interpolation of scattered data in three or more independent variables. It covers schemes that can be used for any number of variables as well as schemes specifically designed for three variables. Emphasis is on breadth rather than depth, but there are explicit illustrations of different techniques used in the solution of multivariate interpolation problems.

List of Contents

§1. Introduction

Multivariate scattered data interpolation problems arise in many different ways, as suggested by the following examples (others can be found in the literature listed at the end of this paper):

1. **Gravitational Field of the Earth [53].** Earth's gravitational field constitutes a vector valued function of three components. It has been measured with great accuracy at many points on the surface of the earth, and in space. Yet no comprehensive model is currently available that is sufficiently accurate for all applications. Major difficulties in modeling the field are caused by the presence and significance of disparate spatial scales: On the surface of the planet the field may change appreciably within a few miles. It is affected—both in strength and direction—e.g., by the presence of mountain ranges. Gravitational effects on a larger scale affect a satellite in a typically low orbit of a few hundred kilometers. Finally, for the purposes of navigating a space craft in the vicinity of another planet, the Earth can be thought of as a point source of gravity.
2. **Color Film Processing [48].** In generating color prints or slides, a processing system has to be corrected for effects, e.g., of film, chemistry, lenses, and filters. The correction is a (vector valued) function of the intensities of three colors (Red, Green, Blue, i.e., RGB) on certain test slides. Appropriate corrections can be determined for specific RGB values. Corrections for other RGB values can then be obtained by interpolation.
3. **Implicitly Defined Surfaces [73].** Tom Sederberg has explored the design and representation of (two-dimensional) surfaces as the contour surfaces of trivariate piecewise polynomial functions. Such functions can be constructed by interpolation. The data then serve as *Design Handles*.
4. **Oil exploration.** In exploratory oil drilling an abundance of data is obtained at each bore hole. The data are distributed in a peculiar fashion: they are dense along bore holes and widely scattered otherwise.

In this paper, we consider the *scattered data interpolation problem:* Given

$$(x_i, y_i) \in \Omega \times \mathbb{R}, \quad i = 1, 2, \cdots, N, \quad \Omega \subset \mathbb{R}^k, \tag{1}$$

find $s \in S(\Omega)$ such that

$$s(x_i) = y_i, \quad i = 1, 2, \cdots, N. \tag{2}$$

The points (x_i, y_i) are the *data points* (or just *data*). We refer to the x_i as *data sites* to separate them from the *function values* y_i. The integer k is the number of independent variables. Our case of interest is

$$k > 2. \tag{3}$$

The integer N is the number of data to be interpolated to, and Ω is a suitable domain containing the data sites. $S(\Omega)$ is the *interpolating space*, i.e., a (usually finite dimensional) linear space of functions defined on Ω. Usually, $S(\Omega) \subset C^1(\Omega)$.

It is often convenient (e.g., for describing an interpolation scheme) to think of the data as having been generated by a *primitive function* f, i.e.,

$$y_i = f(x_i), \quad i = 1, 2, \cdots, N. \tag{4}$$

However, we do not assume that any information about f is available other than the data themselves.

A completely specified procedure of prescribing an interpolation space S and an interpolant s (for general Ω and general data) constitutes an interpolation *scheme*. This is distinguished from a *technique* which is used to build an interpolation scheme. An *algorithm* is a scheme that has been implemented in a commercially or freely available computer program. Emphasis in this paper is on techniques, and not on schemes or algorithms.

To make the scope of this paper manageable, a number of topics and issues had to be omitted:

1. **Approximation Problems** where the *residuals* $e_i = s(x_i) - y_i$ are to be made small in some appropriate sense. For practical applications, this is probably a more important objective than actual interpolation (which renders the residuals zero) since realistic data usually are comprised of measured values and include errors whose exact reproduction is pointless. Sometimes an approximation scheme can be turned into an interpolation scheme by choosing the dimension of the approximating space equal to the number of data. Used frequently, and related to some of the techniques described in this paper, are *smoothing splines* [82], i.e., functions that minimize
$$I(s) = \sum_{i=1}^{N} \big(s(x_i) - y_i\big)^2 + \lambda J(s) \tag{5}$$
where λ is a parameter and $J(s)$ is a suitable semi-norm, measuring for example the strain energy in a clamped elastic plate (see section 4.2.2). There are many additional references on smoothing splines in the bibliography [41].
2. **Interpolation Error.** This is the difference $\|s - f\|$ for some primitive function f (and a suitable norm $\|\cdot\|$), usually considered a function of the density of the data sites. This is an issue in classical approximation theory.
3. **Multivariate B-Splines.** This is a large and fascinating area in its own right. However, to date multivariate B-splines do not appear to have been particularly competent in solving multivariate scattered data interpolation problems. There is a large number of papers on multivariate B-splines in the bibliography [41].
4. **Geometric Continuity.** In Computer Aided Geometric Design (of one and two dimensional surfaces embedded in $\mathbb{R}^3$) a central objective is to recognize, obtain, and represent properties of a surface that are independent of any particular parameterization. This allows a great degree of

flexibility in the choice of the parameterization, and in the structure of the surfaces that can be designed. For higher dimensional surfaces this issue has not yet played a prominent role. The only reference on this topic known to the author is Dieter Lasser's recent thesis [50]. In this paper, we restrict ourselves to the functional case (1) (which of course includes *vector valued* functions defined on Ω).

I have attempted to cover existing techniques as widely as possible. However, only primary references are listed at the end of this paper. Their bibliographies should be consulted for further information. A recent extensive bibliography of 1107 entries on multivariate approximation and interpolation is [41]. Recent collections of papers include [31], [36], and the five volumes of the CAGD journal that have appeared to date. A rich source of information and ideas is the Winter 1984 issue of the *Rocky Mountain Journal of Mathematics* which is entirely devoted to multivariate interpolation and approximation problems.

It is unlikely that multivariate interpolation schemes are going to emerge that are truly *general purpose* in the sense that they will solve satisfactorily the bulk of realistic problems. This applies to problems in three or more variables even more so than the case of just two independent variables. There is a tremendous degree of variety and peculiarity present in multivariate interpolation problems. Thus it is important to be aware of many different techniques of designing interpolation schemes, and of obtaining certain effects. Many such techniques are illustrated explicitly in this paper. The purpose of these illustrations is always the technique itself, and not a detailed description of the particular interpolation scheme to which it is applied.

In spite of the the complexity of real problems, most of the schemes defined in this paper are *general purpose* in the sense that they are designed for generic situations rather than specific applications. (The exceptions are: tensor product schemes, interpolation on the sphere and other surfaces, and hypercubal methods). Calling a method *general purpose* requires certain underlying assumptions that should be made explicit:

1. There is no readily utilizable structure in the data that calls for the use of a more specific interpolation scheme. (In particular, it is hard to conceive of a situation where one would want to interpolate to tensor product data with a scattered data scheme.)
2. On the other hand, there is no *peculiar* structure that would upset a *general purpose method.* For example, a scheme based on a (k-dimensional) triangulation of the data sites is likely to perform poorly in a situation where data arise densely along scattered lines.
3. There is no appreciable noise in the data. (Otherwise one would use an approximation rather than an interpolation scheme.)
4. The representation of the data by a smooth (e.g., C^1) function is appropriate. This is often not the case. For example, in Computer Aided Tomography one is particularly interested in surfaces across which the density of the human body changes discontinuously (i.e., the boundary

of organs and bones). Similarly, in oil exploration a major issue is the location of *faults* (where subterranean layers of rock have been ruptured). The prediction and modeling of unknown discontinuities is an open and difficult problem that most likely will have to be addressed in the context of individual applications.

5. We will always assume that the data are such that the existence and uniqueness of an interpolant is assured. In several variables there are many settings where this is not the case. For example, it is impossible to interpolate to three data by a bivariate linear function if the data sites are situated along a straight line. Similarly, one cannot interpolate with a bivariate quadratic to six points whose sites lie on a conic section. Typically, such situations are *singular* in the sense that existence and uniqueness of the interpolant can be restored by an arbitrarily small perturbation of the data sites. In this context, the opposite of *singular* is *generic*. For all of our interpolation schemes, existence and uniqueness are assured at least generically.

In designing, choosing, or using a multivariate interpolation scheme several issues may have to be considered:

1. How can a function of more than three variables be rendered?
2. Is the scheme *local* or *global*, i.e., is the value of the interpolant at a given point dependent on all data or only a *small* subset of data sites close to the point of evaluation? A precise and general definition of these terms is not possible, but for any particular scheme it usually makes sense to classify it as local or global. Local schemes are more attractive for applications, while global schemes are easier to build and to design so as to possess other desirable properties (which depend on the particular problems that are to be solved).
3. What is the *degree of precision* of a scheme? This is the largest number m such that $s = p$ whenever p is polynomial of degree up to m and $y_i = p(x_i)$, $i = 1, 2, \cdots, N$. The degree of precision is loosely related to the *shape fidelity* of a scheme. No precise statements can be made, but it is hard to envision viable schemes that do not have at least linear precision. On the other hand, a high degree of precision has no value in itself.
4. What is the *degree of smoothness* of the interpolant, i.e., how often is it differentiable? Usually, first order differentiability is required.
5. What kind of data are required for the construction of the interpolant? As a rule, only functional data are provided by the problem, as indicated in (1). However, other data, e.g., values of derivatives at data sites, or function and derivative values at points other than data sites, may be required by the scheme to obtain properties such as localness, or increased precision. See section 11 for a discussion of derivative generation techniques.
6. Is the scheme invariant under affine transformations? Let A be a non-singular $k \times k$ matrix, γ a non-zero real number, and let $\bar{s}$ be the so-

lution of the modified interpolation problem with the data $(Ax_i, \gamma y_i)$, $i = 1, 2, \cdots, N$. The interpolation scheme is *affinely invariant* if, for all data sets, and all points $x \in \Omega$, $\bar{s}(Ax) = \gamma s(x)$. Affine invariance implies for example that interpolated physical quantities are independent of the units chosen for expressing the data.

7. Does the interpolant depend continuously on the data sites? This property is clearly desirable yet it is not present e.g., for any triangulation based scheme since the triangulation itself depends discontinuously on the data sites.
8. A variety of properties may be desirable depending on the particular interpolation problem. For example, the value of the interpolant may be constrained to be bracketed by the minimum and maximum function values; the interpolant may have to be convex, or monotone with respect to some or all of the independent variables; or it may be required to minimize a physically relevant variational principle over an affine space of potential interpolants.

We conclude this section by introducing a few terms and some notation that will be used throughout the paper:

When it is necessary we will denote the components of $x \in \mathbb{R}^k$ by ξ_i, $i = 1, 2, \cdots, k$, to avoid confusion with the data sites. The *norm* $\|\cdot\|$ will be the *Euclidean Norm*, i.e.,

$$\|x\| = \sqrt{\sum_{i=1}^{k} \xi_i^2}. \tag{6}$$

The expression $\mathbb{P}_d^k$ denotes the $\binom{d+k}{k}$-dimensional linear space of polynomials of k variables of total degree d.

The interpolant s will be sought in the finite dimensional linear interpolation space

$$S(\Omega) = \text{span}\left\{w_i \in C^1(\Omega) \colon i = 1, 2, \cdots, M\right\}. \tag{7}$$

where $M \geq N$. Thus it will be of the form

$$s(x) = \sum_{i=1}^{M} \alpha_i w_i(x) \tag{8}$$

where the α_i are *coefficients* that are determined by the interpolation conditions (2) and, if $M > N$, by additional auxiliary conditions that are imposed to enforce certain useful properties of s. The w_i are the *basis functions*.

We define the *Kronecker Delta* as usual by

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}. \tag{9}$$

A frequently arising special case is given when $M = N$ and

$$w_i(x_j) = \delta_{ij} \tag{10}$$

In that case,

$$s(x) = \sum_{i=1}^{N} y_i w_i(x) \tag{11}$$

and the interpolant is said to be in *cardinal form.*

The generalization of a triangle to k-dimensional space is a *k-simplex* or just a *simplex.* A simplex $\mathcal{S}$ is the convex hull of $k+1$ points called the *vertices* of the simplex. $\mathcal{S}$ is *non-degenerate* if its k-dimensional volume is non-zero, and *degenerate* otherwise. The convex hull of any subset of $\kappa+1$ vertices of a simplex is called a *(κ-)face* of the simplex. A face that is the convex hull of precisely k vertices is also called a *facet.* A 1-face is also called an *edge.* Note that a κ-face of a simplex is itself a κ-dimensional simplex. (In [7] and [84], the meanings of the terms *facet* and *face* are interchanged. On the other hand, the usage proposed here is also employed e.g., in [27] and [52]. There seem to be no linguistic reasons to prefer one usage over the other.)

The location of a point $x \in \mathbb{R}^k$ can be expressed uniquely in terms of *barycentric coordinates* b_i with respect to a non-degenerate simplex $\mathcal{S}$ with vertices V_i, $i = 1, 2, \cdots, k+1$ as

$$x = \sum_{i=1}^{k+1} b_i V_i \quad \text{where} \quad \sum_{i=1}^{k+1} b_i = 1. \tag{12}$$

A polynomial p of degree d can be expressed uniquely in *Bézier-Bernstein* form as

$$p(x) = \sum_{i_1+i_2+\cdots+i_{k+1}=d} c_{i_1 i_2 \cdots i_{k+1}} \frac{d!}{i_1! i_2! \cdots i_{k+1}!} b_1^{i_1} b_2^{i_2} \cdots b_{k+1}^{i_{k+1}}. \tag{13}$$

The Bézier-Bernstein form of a polynomial has many useful properties and is virtually indispensable in the design of piecewise polynomial multivariate interpolation schemes. A recent survey of the properties of the Bézier-Bernstein form in a general number of variables is [27].

§2. Rendering of Trivariate Functions

The graphs of bivariate functions are frequently displayed as contour plots, three dimensional line drawings with or without hidden line elimination, or as shaded surfaces. Such surfaces reside in $\mathbb{R}^3$ and can be processed by the human mind as the familiar concept of a three dimensional object projected onto the two-dimensional retina. There is a fundamental biologic limit to similar renderings of higher dimensional objects. This seems to be the reason that pictorial displays of functions of more than two variables do not play nearly the same prominent role as for functions of two variables. However, some efforts have been made to render functions s of precisely *three variables.* Two obvious approaches suggest themselves: 1. Display a number of *cross* *sections* where one of the variables is held constant, or, 2. display *contour surfaces* where the value of $s(x)$ equals some constant.

The latter approach is illustrated e.g., in [16]. Particular schemes of displaying contour surfaces of a function s are described in [61] and [74]. Petersen *et al* assume that s is piecewise polynomial (of any degree) in Bézier-Bernstein form on a tetrahedral *tessellation* (but not necessarily a *triangulation*) of Ω. Their scheme comes with an interpolation scheme of this character which is described in section 8.1. The contouring scheme proceeds by approximating a polynomial p on a given tetrahedron t by another polynomial of degree one less than that of p. If the approximation is sufficiently accurate it is accepted. Otherwise t is split (about its longest side) into two subtetrahedra t_1 and t_2, p is restricted to each of t_1 and t_2, and the process is repeated on both t_1 and t_2. Eventually the approximation becomes a continuous piecewise linear function on a tetrahedral tessellation. The corresponding planar segments are then drawn to generate a *continuous piecewise linear* contour surface. The scheme has the advantage of spending computational effort only where it is needed, i.e., it is *adaptive*. In an alternative approach, Sewell assumes that function values are given on a uniform three dimensional rectangular grid. (Of course, any interpolating function can be evaluated on such a grid.) The contour surfaces are drawn as frame works of overlapping *bands* (i.e., opaque lines of non-zero thickness). Bands in the background can be seen through gaps between bands in the foreground. Sewell's scheme has the advantage of being available as a FORTRAN algorithm [75].

§3. Tensor Product Schemes

This section comprises an aberation from the main theme of this paper in that it assumes a very special distribution of the data sites. Suppose for simplicity that $k = 3$. Deviating momentarily from the notation defined in (1), suppose the data are of the form $(\zeta_i, \eta_j, \theta_k)$, $\mu = 1, \cdots, N_\zeta$, $\nu = 1, \cdots, N_\eta$, $\kappa = 1, \cdots, N_\theta$, and we wish to find an interpolant of the form

$$s(\zeta, \eta, \theta) = \sum_{\mu=1}^{N_\zeta} \sum_{\nu=1}^{N_\eta} \sum_{\kappa=1}^{N_\theta} \alpha_{\mu\nu\kappa} p_\mu(\zeta) q_\nu(\eta) r_\kappa(\theta) \tag{14}$$

satisfying the interpolation conditions

$$s(\zeta_\mu, \eta_\nu, \theta_\kappa) = y_{\mu\nu\kappa}, \quad \mu = 1, \cdots, N_\zeta, \nu = 1, \cdots, N_\eta, \kappa = 1, \cdots, N_\theta. \tag{15}$$

This is a *tensor product* problem. It can be reduced to a set of *univariate* interpolation problems. Suppose for simplicity that the basis functions for each variable are given in *cardinal form*, i.e.,

$$p_i(\zeta_j) = \delta_{ij}, \quad q_i(\eta_j) = \delta_{ij}, \quad r_i(\theta_j) = \delta_{ij}. \tag{16}$$

The interpolant can then be written simply as

$$s(\zeta, \eta, \theta) = \sum_{\mu=1}^{N_\zeta} \sum_{\nu=1}^{N_\eta} \sum_{\kappa=1}^{N_\theta} y_{\mu\nu\kappa} p_\mu(\zeta) q_\nu(\eta) r_\kappa(\theta). \tag{17}$$

Any set of (univariate) non-cardinal basis functions can be converted to cardinal form by calculating the inverse of the corresponding Vandermonde matrix, e.g., the matrix $[p_i(\zeta_j)]_{i,j=1,\cdots,N_\zeta}$ for the variable ζ. In this manner, the effort of finding the coefficients of the interpolant is reduced from $O\big((N_\zeta N_\eta N_\theta)^3\big)$ to $O\Big(N_\zeta^3 + N_\eta^3 + N_\theta^3\Big)$ which constitutes a substantial saving. Tensor products do not have to be implemented in cardinal form, but in any case they can be reduced to univariate rather than truly multivariate interpolation problems. More detailed discussions of tensor product schemes are in [25] and [26]. See also [55] for a more abstract discussion of tensor product approximation.

§4. Point Schemes

The term *Point Schemes* refers to interpolation schemes that are not based on a tessellation of the underlying domain Ω.

4.1 Shepard's Methods

Shepard's method [76] may be the best known among all scattered data interpolants in a general number of variables. In its simplest form, it is given by

$$s(x) = \sum_{i=1}^{N} w_i(x) f(x_i) \quad \text{where} \quad w_i(x) = \frac{\|x - x_i\|^{-p}}{\sum_{j=1}^{N} \|x_i - x_j\|^{-p}}. \tag{18}$$

In this form, the evaluation of s at a data site leads to a division by zero. However, the definition of s can be extended continuously by the interpolation requirement $s(x_i) = y_i$.

Shepard's method is a particular example of a *convex combination* based scheme, i.e., the weights w_i are non-negative and sum to 1. The interpolant is arbitrarily often differentiable if $p > 1$. The most frequent choice is $p = 2$, in which case the basis functions w_i are *rational*. If $p > 1$, the first derivatives vanish at the data points, i.e., the interpolant has *flat spots*. The scheme is obviously global and has only constant precision. It does have the property

$$\min_{i=1,\cdots,N} \{y_i\} \le s(x) \le \max_{i=1,\cdots,N} \{y_i\} \quad \forall x \in \mathbb{R}^k \tag{19}$$

which is sometimes required. However, for most purposes Shepard's method in its unmodified form yields unacceptable interpolants. This is perhaps best illustrated by the fact that it has never been proposed as an interpolation scheme in the *univariate* case where many other techniques are readily available.

The deficiencies of Shepard's method can be overcome in various ways. We describe some of them:

1. **Mollifying Basis Functions.** To localize Shepard's method one can multiply the basis functions w_i by *mollifying functions* μ_i satisfying

$$\mu_i(x_i) = 1 \tag{20}$$

and having local support in some appropriate sense. For example, the *Franke-Little weights* [22] are given by

$$\mu_i(x) = \left(1 - \frac{\|x - x_i\|}{R_i}\right)_+^\nu \tag{21}$$

where

$$x_+^\nu = \begin{cases} x^\nu & \text{if } x > 0 \\ 0 & \text{if } x \le 0 \end{cases}, \tag{22}$$

the R_i are suitably chosen radii of circles around the data sites that constitute the support of the modified basis function, The smoothness of the multiplying factor is $\nu - 1$. This technique yields an interpolant

$$s(x) = \sum_{i=1}^{N} y_i \mu_i(x) w_i(x) \tag{23}$$

which is still in cardinal form. However, the (constant) precision of the scheme has been destroyed.

2. **Interpolation to Taylor operators.** To increase the precision of Shepard's method one can interpolate to Taylor expansions instead of function values. For example, the interpolant defined by

$$s(x) = \sum_{i=1}^{N} w_i(x) \left(f(x_i) + \nabla f(x_i)^T (x - x_i)\right) \tag{24}$$

has linear precision (since the weights w_i add to 1 and the individual Taylor operators have linear precision). Since $\nabla w_i(x_j) = 0$ for all i and j it also follows that $S_p^1 f$ interpolates to $\nabla f(x_i)$. The precision can be increased further by interpolating to Taylor operators of order greater than linear. However, for Shepard's method, if higher order Taylor operators were to be used, higher order derivatives would not be interpolated to. Obviously, the technique calls for additional derivative values that are not normally available as data.

3. **Boolean Sums.** Let P and Q be two linear operators defined on a suitable function space, and such that the composition PQ is defined. The *Boolean Sum* of P and Q is defined by

$$S = P \oplus Q = P + Q - PQ \tag{25}$$

Barnhill and Gregory show in [18] that S has (at least) the interpolation properties of P and the precision properties of Q. Thus one can obtain an interpolation scheme with an arbitrarily high degree of precision m, say, by letting Pf be Shepard's interpolant and $Qf \in \mathbb{P}_m^k$ be a Least Squares Approximation.

4. **Delta Sums [38], [39].** A Boolean Sum can be formed only if the operator Q can be applied when f is represented solely by the given

data. This fact rules out, e.g., tensor product based operators. This motivated Foley to define a *Delta Sum* of two operators P and Q as

$$S = P \bigtriangleup Q = P \oplus (QP) = QP + P(I - QP).$$

The process can be repeated to yield a *Delta Iteration.* Several Boolean Sum and Delta Sum based interpolation schemes are described and discussed in [22].

A recent paper on Shepard's method is [37]. It surveys the properties of the method, and in particular analyzes its approximation order. Other papers on implementations, modifications, and tests of Shepard's method include [15], [17], [40], [42], [49], and [56].

4.2 Radial Interpolants

The term *radial* is due to Rippa [68]. Radial interpolants are of the form

$$s(x) = \sum_{i=1}^{N} \alpha_i g(\|x - x_i\|) + p_m(x) \tag{26}$$

where g is a given univariate so-called *radial function*, and $p_m \in \mathbb{P}_m^k$. The coefficients of s are determined by the interpolation condition (2) and the additional requirement that

$$\sum_{i=1}^{N} \alpha_i q(x_i) = 0 \tag{27}$$

for all polynomials $q \in \mathbb{P}_m^k$. Obviously, if $y_i = p(x_i)$, $i = 1, 2, \cdots, N$ for some polynomial $p \in \mathbb{P}_m^k$, then the choice $\alpha_i = 0$, $i = 1, 2, \cdots, N$ and $p_m = p$ will satisfy all requirements. Thus radial interpolants have degree of precision m.

Recent discussions of radial schemes are in [35], [57], and [62]. Micchelli's paper establishes in particular that for many radial schemes, including those described here, the interpolant exists and is unique for all data sets (provided only that the interpolation problem with $\mathbb{P}_m^k$ as interpolation space has at most one solution for all sets of function values). Remarkably, this fact was established only after Multiquadrics had been used with great success for fifteen years. Dyn and her co-workers in [35] and some of the references listed there develop particular radial schemes and numerical techniques to deal with the large and ill-conditioned linear system defining the interpolants.

The following subsections describe two specific choices of the radial function g.

4.2.1 Hardy Multiquadrics

These interpolants are defined by

$$g(t) = (h^2 + t^2)^\alpha, \quad \alpha = \pm\frac{1}{2}, \quad h > 0, \quad m = 0 \tag{28}$$

and were first proposed by Hardy in 1971 [44]. The parameter h is at the user's disposal and its best choice is related to the distance between data points. Multiquadrics constitute one of the most successful and widely used general purpose interpolants for multivariate data. Until recently, that success was based entirely on numerical experiments. Further insight into the effectiveness of Multiquadrics was gained by Buhmann [28], [29], and Jackson [47] who study the approximation properties of Multiquadrics.

4.2.2 Duchon Thin Plate Splines

These interpolants were originally introduced by Duchon [34] for the case $k = 2$ as solutions of the variational principle

$$I_\mu s = \int_{\mathbb{R}^k} \|D^\mu(s)\|^2 \mathrm{d}x = \min, \tag{29}$$

where

$$D^\mu = \left(\frac{\partial^\mu}{\partial\xi_{i_1}\partial\xi_{i_2}\cdots\partial\xi_{i_\mu}} : \quad (i_1, i_2, \cdots, i_\mu) \in [1, d]^\mu \right), \quad \mu = m + 1 \tag{30}$$

over all admissible functions, subject to the interpolation conditions (2). In the special case $\mu = k = 2$,

$$I_2 s = \int_{\mathbb{R}^2} \left(s^2_{\xi_1\xi_1} + 2s^2_{\xi_1\xi_2} + s^2_{\xi_2\xi_2}\right)\mathrm{d}\xi_1\mathrm{d}\xi_2 \tag{31}$$

and the functional I_2 measures the strain energy in a clamped elastic plate, giving rise to the name *thin plate spline*.

The solution of the minimization problem (29) are radial interpolants, called *surface splines*, with

$$g(t) = \begin{cases} t^{(2\mu-k)} \log t & \text{if } k \text{ is even} \\ t^{(2\mu-k)} & \text{if } k \text{ is odd} \end{cases} . \tag{32}$$

§5. Natural Neighbor Interpolation

This scheme was introduced by Sibson [77]. It is based on the *Dirichlet* (or *Thiessen* or *Voronoi*) *Tessellation* of Ω. This is the collection of *tiles* of the form

$$\tau_i = \Big\{x \in \mathbb{R}^k \colon \|x - x_i\| \le \|x - x_j\| \quad \forall j = 1, 2, \cdots, N\Big\} \cap \Omega \tag{33}$$

Obviously,

$$\Omega = \bigcup_{i=1}^{N} \tau_i \tag{34}$$

and the tiles are disjoint except for parts of their boundaries.

We assume that the domain $\Omega \subset \mathbb{R}^k$ has finite volume. In order to evaluate the interpolant at a point $x \in \Omega$ we can think of adding x to the set of data sites and carving its tile from the tiles of neighboring data sites. Thus we let

$$\tau(x) = \Big\{z \in \mathbb{R}^k \colon \|x - x_i\| \le \|z - x\| \quad \forall j = 1, 2, \cdots, N\Big\}, \tag{35}$$

$$\tau_i(x) = \tau(x) \cap \tau_i \tag{36}$$

and

$$v(\tau) = \text{volume}(\tau) \tag{37}$$

Sibson's natural neighbor interpolant is then defined as

$$s(x) = \frac{\sum\limits_{i=1}^{N} \lambda_i(x) \|x - x_i\|^{-1} \xi_i(x)}{\sum\limits_{i=1}^{N} \lambda_i(x) \|x - x_i\|^{-1}} \tag{38}$$

where

$$\begin{aligned} \xi_i(x) &= f(x_i) + \nabla f(x_i)^T (x - x_i) \\ \lambda_i(x) &= \frac{v\big(\tau_i(x)\big)}{v\big(\tau(x)\big)} \end{aligned} \tag{39}$$

Note that if we replace $\lambda_i(x)$ by $\|x - x_i\|^{-1}$ we obtain a linearly precise version of Shepard's method. However, $\lambda_i(x) \neq 0$ only for points close to x which causes the scheme to be local. As for Shepard's method, we have to define explicitly $s(x_i) = y_i$.

The natural neighbor interpolant has some remarkable properties:

1. It has linear precision. In fact it also reproduces *spherical quadratics* (i.e., quadratic functions whose matrix of second derivatives is a multiple of the identity matrix).
2. It is local.
3. In the special case $k = 1$ the function s is piecewise cubic.

4. $s \in C^1(\Omega)$
5. The value $s(x)$ depends continuously on the data sites x_i.

The proof of the last two properties is involved.

Obviously, in order to construct the interpolant (38), gradient values have to be supplied. Sibson's paper describes a scheme in the spirit of natural neighbor interpolation that generates derivative values from given function values.

§6. k-dimensional Triangulations

Many bivariate interpolation schemes are based on triangulations of the data set, and it is natural to use the same ideas in more than two variables. Let $\mathcal{T} = \{t_i : i = 1, 2, \cdots, T\}$ be a set of non-degenerate simplices, and let $\Omega = \bigcup_{i=1}^{T} t_i$. It is useful to allow for the possibility that Ω is not convex. The following definition is a slight variation of the definition in [52] (where Ω is the convex hull of the data sites):

Denote the set of data sites by $\mathcal{X} = \{x_i : i = 1, 2, \cdots, N\}$, Then the set $\mathcal{T}$ is a *triangulation* of $\mathcal{X}$ if the following conditions are satisfied:

1. All vertices of each simplex are members of $\mathcal{X}$.
2. The interiors of the simplices are pairwise disjoint.
3. Each facet of a simplex is either on the boundary of Ω or else is a common facet of exactly two simplices.
4. Each simplex contains no points of $\mathcal{X}$ other than its vertices.
5. Ω is homeomorphic to $[0, 1]^k$.

The last of the above conditions rules out degenerate triangulations consisting for example of two tetrahedra touching in just one vertex or edge.

There are some significant differences between bivariate and higher dimensional triangulations. These include for example:

1. Specification of $\mathcal{X}$ and Ω does not determine the number of simplices. Counter examples are given in [19] and [52].
2. If $k > 3$, possible triangulations may not be distinguishable by information about connectivity of points, i.e., by the specification of edges. For examples see [52].
3. A standard technique of building *2-dimensional* triangulations consists of adding one triangle at a time, maintaining at each stage a valid triangulation. For some triangulations, this is not possible if $k > 2$. A simple counter example (consisting of 14 points and 41 tetrahedra) is given in [69], a more complicated one for many different triangulations of a particular domain is given in [24] and also described in [70], p. 17. If the construction is possible the triangulation is said to be *shellable*.

Many criteria have been proposed for selecting particular *2-dimensional* triangulations. Perhaps the easiest to generalize is that of the *Delaunay triangulation*. This is the dual of the Thiessen tessellation: two points are connected if their tiles share a line segment. An equivalent requirement is that for any two neighboring triangles the circumcircle of one triangle does not contain the opposite vertex of the other triangle. This is the *local circle test*. It implies the stronger *global circle test*: the circumcircle of any triangle in the triangulation does not contain any other data site at all. (We ignore special cases where four or more points lie on the circumference of a circle.)

Lawson [52] generalizes these concepts. He shows:

1. A set of $k+2$ points in $\mathbb{R}^k$ may be triangulated in at most 2 different ways.
2. The (local) sphere test (the obvious generalization of the local circle test) selects a preferred one of these two triangulations.
3. A triangulation that satisfies the local sphere test also satisfies the global sphere test.
4. A triangulation satisfying the global sphere test is dual to the Thiessen tessellation, i.e., the tiles of two points connected by an edge share a $k-1$ dimensional boundary polygon (again we consider only the generic case).

Pascal codes for constructing k-dimensional triangulations are given in [45]. They contain facilities for forcing certain facets to be contained in the triangulation. This option can be used e.g., to generate non-convexities, holes, and cavities.

We now give a precise definition of the localness of triangulation based schemes that is suitable for the purposes of this paper. More specifically, we generalize the term *minimally supported* defined in [13] for triangular schemes.

The *star* of a κ-face φ is the union of all simplices that contain φ. A function s defined on a k-dimensional triangulation is said to be *minimally supported* if there is a κ-face φ in the triangulation such that the support of s is contained in the star of φ. A linear space $S(\Omega)$ is *minimally supported* if it has a basis consisting of minimally supported functions. An interpolation scheme is *minimally supported* if the interpolation space S is minimally supported.

Note that in spite of the term's negative connotation, it is highly desirable for a space or a scheme to be minimally supported.

If $S \subset C(\Omega)$ contains all constant functions, then for any face φ there must be a function $s \in S$ whose support comprises all simplices containing φ. Otherwise, all functions $s \in S$ would have to vanish on φ. This fact gave rise to the term *minimally* supported. It is reasonable to consider a minimally supported scheme to be local. On the other hand, a scheme might also be called *local* without being minimally supported. It is, however, no easy task to find a reasonable more general definition of the word *local*. For example, in the case of of 2-dimensional triangulations an easy fallacy would be to define a scheme to be local if for each basis function w there exists a vertex V such that the support of w consists of the union of the stars of all vertices contained in the star of V. However, simple examples show that such a support set may not even be simply connected, which renders the utility of the definition doubtful. The problem is compounded for triangulations of dimension greater than 2.

§7. Tetrahedral Schemes

In this chapter, we consider schemes that have been specifically designed for interpolants on a *three-dimensional triangulation*. Interpolants that apply to triangulations of a general dimension are described in the next chapter.

The original motivation for defining interpolants piecewise on tetrahedra is the prospect of obtaining local schemes. Indeed, all schemes described in this chapter are minimally supported. However, with the increasing availability of *parallel processing* tetrahedral (and, more generally, simplicial) schemes are likely to become even more attractive since individual tetrahedra can be processed independently and simultaneously!

7.1 Polynomial Schemes

It is natural to consider the use of finite elements as an interpolant, particularly since there is a large and sophisticated machinery available for handling them. If a piecewise defined interpolant is to be polynomial on each tetrahedron and globally differentiable, then its polynomial degree must be at least 9 [86]. Rescorla [67] gives an explicit description of such a scheme. It requires 220 (i.e., the dimension of the space of trivariate nonic polynomials) data per tetrahedron. These include derivatives through fourth order at the vertices of the tetrahedron, as well as function and derivative values at various points on the faces and the centroid of the tetrahedron. The degree of precision of the scheme is nine. Rescorla's approach consists of constructing a basis of $\mathbb{P}_9^3$ (in terms of barycentric coordinates) that renders triangular the Vandermonde matrix associated with the data.

In the bivariate case, the minimal polynomial degree for a piecewise polynomial C^1 scheme is five. A well-known technique of reducing that degree, as well as the degree of the derivative values required as data, is to subdivide the triangle into subtriangles. Splitting a triangle about its centroid into three subtriangles and letting the polynomial degree be 3 gives rise to the widely used Clough-Tocher scheme [3], [80], which requires only first order derivative data.

A similar approach to the trivariate problem is taken in [3]. The tetrahedron is split about its centroid into 4 subtetrahedra, the interpolant is *quintic* on each subtetrahedron, and the degree of the required derivative data is 2 (as opposed to 4 required for the nonic polynomial finite element). The scheme is such that perpendicular first order cross-boundary derivatives across a face of the tetrahedron have only quadratic precision. The overall degree of precision is therefore only cubic. A peculiarity of the scheme is that four of its coefficients are not needed for either interpolation or forcing global differentiability, and have to be *condensed*, i.e., conditions have to be made up that eliminate them. The interpolant is given in Bézier-Bernstein Form. The MRC report [4] lists explicit expressions for the coefficients of the scheme.

An alternative approach is described in [85]. The bivariate Powell-Sabin split [63] is applied to each facet of the tetrahedron and all points obtained in this manner on the boundary of the tetrahedron are connected to a suitably chosen interior point. The interior points chosen on the facets need not be the centroids. The interpolant is *quadratic* on each of the resulting 24 subtetrahedra. (This low polynomial degree was originally motivated by the ease of contouring quadratic functions.) In order for the overall interpolant to be globally differentiable on a 3-dimensional triangulation, the interior points of the facets and the tetrahedra must satisfy certain geometric constraints. It is an open question whether the constraints can be satisfied in general. In the special case that the points about which edges, facets, and tetrahedra are split can be chosen to be the circumcenters of the corresponding faces, the constraints are satisfied. However, the circumcenters may not be interior points of their face. Worsey and Piper give a criterion for a triangulation to be such that all circumcenters are strictly interior.

7.2 Rational Schemes

Piecewise polynomial functions are usually preferred for applications because they are particularly easy to evaluate, differentiate, and integrate. However, they are *not* particularly easy to piece together smoothly. A great deal of flexibility can be gained by using *rational* functions.

A particularly serviceable class of basic interpolation operators on triangles and tetrahedra are *Barnhill-Birkhoff-Gordon (BBG)* projectors [2], [19]. Here we discuss their application to a tetrahedron t. We denote the *edges* of t by

$$e_{ij} = V_j - V_i, \quad i, j \in \{1, 2, 3, 4\}, \quad i \neq j \tag{40}$$

where the V_j are the vertices of t. There is a BBG projector associated with each edge. For simplicity of notation, we describe only the projector associated with e_{12}. For a general point $x \in t$ we consider the line parallel to e_{12} through x. That line intersects the facet $b_1 = 0$ in a point X_1 and the facet $b_2 = 0$ in a point X_2. Then we interpolate to function and derivative values at X_1 and X_2 and evaluate the so obtained *univariate* interpolant at x. Specifically, we define

$$\begin{aligned} P_{12}f = \quad & h_0\left(\frac{b_2}{b_1+b_2}\right)f(X_2) \\ & + h_1\left(\frac{b_2}{b_1+b_2}\right)f(X_1) \\ & + \bar{h}_0\left(\frac{b_2}{b_1+b_2}\right)(b_1+b_2)\frac{\partial f}{\partial(e_{12})}(X_2) \\ & + \bar{h}_1\left(\frac{b_2}{b_1+b_2}\right)(b_1+b_2)\frac{\partial f}{\partial(e_{12})}(X_1) \end{aligned} \tag{41}$$

where

$$X_1 = (b_1 + b_2)V_2 + b_3V_3 + b_4V_4, \tag{42}$$

$$X_2 = (b_1 + b_2)V_1 + b_3V_3 + b_4V_4, \tag{43}$$

$$h_i, \ \bar{h}_i \in \mathbb{P}_3^1, \quad i \in \{0, 1\} \tag{44}$$

$$h_i(j) = \delta_{ij},\ h_i'(j) = 0, \quad \bar{h}_i(j) = 0,\ \bar{h}_i'(j) = \delta_{ij}, \quad i, j \in \{0, 1\} \tag{45}$$

and the barycentric coordinates b_i are defined in (12). The *directional derivatives* in (41) are defined for general directions $e \in \mathbb{R}^k$ by

$$\frac{\partial f}{\partial e}(x) = \frac{\mathrm{d}}{\mathrm{d}t} f(x + te)\big|_{t=0} \tag{46}.$$

Note that these are *unnormalized* derivatives. Indeed, in dealing with a tetrahedron the notion of *unit* length is not as appropriate as the concept of an edge providing a suitable length scale. If the gradient of the function f is defined then

$$\frac{\partial f}{\partial e}(x) = \nabla f(x) \circ e. \tag{47}$$

(Note that there is no conceptual difficulty in differentiating in the direction of the zero vector, $\partial f / \partial 0 = 0$.) Particularly simple are the edge derivatives of the barycentric coordinates:

$$\frac{\partial b_i}{\partial e_{jk}} = \delta_{ik} - \delta_{ij} \tag{48}$$

Using these facts, it is easy to verify that

$$b_1 = 0 \text{ or } b_2 = 0 \quad \Longrightarrow \quad P_{12}f(x) = f(x) \text{ and } DP_{12}f(x) = Df(x) \tag{49}$$

where D denotes any first order directional derivative. Thus $P_{12}f$ interpolates to function values and first order derivatives *everywhere* on the two facets not containing e_{12}. It is a *transfinite* interpolant.

The data required for a transfinite interpolant will not usually be available. However, the transfinite data are required on a *lower dimensional* face of the tetrahedron and can be *replaced* with a lower dimensional interpolant requiring lower dimensional data. This process is called *discretization.* For example, in (41) one might replace $f(X_1)$ by its linear interpolant

$$f\big((b_1 + b_2)V_2 + b_3 V_3 + b_4 V_4\big) \longleftarrow (b_1 + b_2) f(V_2) + b_3 f(V_3) + b_4 (V_4)\,. \tag{50}$$

Of course, in discretizing a transfinite interpolant one has to incorporate requirements such as global smoothness or a certain degree of polynomial precision. The above simple linear discretization would preclude any degree of precision greater than 1.

The operator P_{12} discussed so far yields an interpolant on only two facets of the tetrahedron. In order to obtain an interpolant on the entire tetrahedron one has to combine several operators. This can be done in several ways. We discuss two of them, each leading to minimally supported discrete C^1 schemes.

1. Boolean Sums. In [19], Barnhill and Little propose the Boolean Sum

$$Qf = P_{12} \oplus P_{34} f. \tag{51}$$

The resulting scheme requires some second order derivative values as data and implies the *(twist) compatibility condition*

$$\frac{\partial^2 f}{\partial e_{12}\partial e_{34}}(x) = \frac{\partial^2 f}{\partial e_{34}\partial e_{12}}(x) \quad \forall x \in e_{13} \cup e_{14} \cup e_{23} \cup e_{24}. \tag{52}$$

This condition would be satisfied if we were actually given a sufficiently smooth primitive function f, but f is used only as a conceptual vehicle towards obtaining a final discrete interpolant. The discretizations on two different facets sharing an edge will usually lead to a violation of the compatibility condition along the edge. However, this drawback can be overcome by adding a sophisticated rational correction term. The compatibility correction and the discretization are discussed at length in [19].

2. **Convex Combinations.** In [2], the transfinite interpolant

$$Sf = a_{12}P_{12}f + a_{34}P_{34}f \tag{53}$$

is proposed where

$$a_{12} = \frac{b_3^2 b_4^2}{b_1^2 b_2^2 + b_3^2 b_4^2}, \quad a_{34} = 1 - a_{12}. \tag{54}$$

The scheme does not necessitate compatibility corrections, and can be discretized to result in a scheme requiring only function and gradient values at the vertices and having quadratic precision. Note that in (53) *discontinuous* blending functions a_{12} and a_{34} are used to yield a globally *differentiable* interpolant.

BBG projectors can be defined for higher degrees of smoothness than C^1. In [11] a *bivariate* C^2 scheme based on the Boolean Sum of three BBG projectors is derived and discretized. In the trivariate case, however, the algebraic manipulations and the codes become too cumbersome to be useful [2], [7]. In section 8.2, an alternative technique for obtaining schemes of higher dimension and higher degrees of smoothness is described.

§8. Simplicial Schemes

In this chapter, we discuss methods that are defined on k-dimensional triangulations of the data set. As for tetrahedral schemes, all of the schemes are minimally supported.

8.1 Polynomial Schemes

Finite element type piecewise polynomial schemes can be defined recursively on simplices of a general dimension [50]. The necessary polynomial degree n increases exponentially with k. As in the bivariate and trivariate case, n can be decreased by subdividing each simplex.

The latter approach is taken by Worsey and Farin [84] who propose a scheme with several remarkable properties: it is piecewise cubic, applicable to any k-dimensional triangulation, requires only function and first order derivative values, and has cubic precision. The key to obtaining the scheme is a particular triangulation of a k-simplex. More generally, the triangulation of a κ-simplex φ ($\kappa \leq k$) is defined recursively as follows:

1. If $\kappa = 1$, φ is not split. (In this case, φ is an edge.)
2. If $\kappa > 1$ then first all faces of φ are split. Then an interior point P of φ is chosen and connected by a line segment to all boundary points of φ and all previously chosen interior points on the faces of φ.

Thus a k-simplex is split into $(k+1)!/2$ subsimplices. In the case $k = 2$ one obtains the Clough-Tocher split, except that the interior point need not be the centroid. If $k > 2$, the choice of the interior points of the various simplices and faces must satisfy the following geometric requirements:

1. Common faces between neighboring simplices are split identically on each simplex.
2. When splitting any two simplices sharing a facet in the triangulation, the interior points chosen in each must be collinear with the interior point chosen to split the common face.

Worsey and Farin [84] show that these objectives can indeed be accomplished. On the resulting split they construct their interpolant as a generalization of the bivariate Clough-Tocher split [80].

8.2 Rational Schemes

The most general discrete simplicial interpolation schemes are *perpendicular interpolants* described in [7]. The resulting interpolants are globally m times differentiable and have polynomial degree of precision m or $m + 1$, depending on the variant that is used. The parameter $m \geq 0$ can be chosen to suit the application. The only data requirements are function values and derivatives through m-th order at the vertices of the k-dimensional triangulation. Because of their intrinsic interest, and because their design illustrates several useful principles, we will discuss them in some detail.

A major difficulty in designing smooth local simplicial schemes is to enforce a certain degree of global differentiability without making reference to the geometry of neighboring simplices (otherwise the scheme would not be local). One therefore typically considers derivatives in directions *perpendicular* to a facet and requires that those derivatives are determined *uniquely* by the data given on the facet. Since neighboring simplices share the facet, the derivatives on the individual simplices are determined identically and agree on that facet, i.e., they are continuous. The reason for choosing a perpendicular cross-direction is because it must be across the boundary and it must be chosen identically on neighboring simplices without knowledge of the geometric configuration of those simplices. This strategy is also used regularly in the finite element technique.

However, directions perpendicular to facets are intrinsically arbitrary as far as the geometry of the local simplex is concerned. A much more natural set of directions is given by the edges of the simplex. Thus many simplicial interpolation schemes (such as the BBG schemes discussed above) are designed in terms of derivatives in the direction of edges, and global smoothness is enforced almost as an afterthought by expressing perpendicular cross-boundary derivatives in terms of edge derivatives. This makes the schemes exceedingly unwieldy, and is the sole reason that e.g., the rational tetrahedral C^1 scheme [2] could not be modified to obtain a C^2 scheme.

The key idea in [7] is to incorporate perpendicular cross-boundary derivatives *directly*. Let $\mathcal{S}$ be the simplex in question, and let F be a facet of $\mathcal{S}$. We think of all faces of $\mathcal{S}$ as embedded in low dimensional affine subspaces, so that any line can intersect a face even though the line may not intersect the simplex $\mathcal{S}$ itself. The *anchor* of F is that facet other than F that forms the smallest angle with F. With a point x and the facet F we associate the straight line through x perpendicular to F. This is the *line of fixation* of x with respect to F. F is also called the *base face* of the line of fixation. The point $B_F(x)$ where the line of fixation intersects its base face is the base point of x and the point $A_F(x)$ where it intersects the anchor is the *top point* of x (all with respect to the base face). Note that the base point as well as the top point may lie outside the simplex $\mathcal{S}$.

Now, for any given point x, we obtain the value of the interpolant $s(x)$ by interpolating along each line of fixation through x to function values and derivative values through m-th order perpendicularly to the base face at the base point. We may also interpolate to the function value at the top point. The $k+1$ values so obtained (one for each facet of S) are then blended together by forming a suitable convex combination. The resulting scheme is *transfinite*. It is discretized by replacing the transfinite data on faces of the simplex by lower-dimensional perpendicular interpolants. The scheme is thus defined recursively. Derivatives are interpolated to by less smooth interpolants. For the special case that a continuous function is required we use linear interpolation, and for the special case that the dimension of the face is 1 we use polynomial Hermite interpolation of the appropriate degree.

For simplicity, we illustrate only the *unanchored* case where no interpolation takes place at the anchor. The interpolants are defined by

$$P_{\mathcal{S}}^m(f)(x) = \begin{cases} \sum\limits_{\mathcal{F}} c_{\mathcal{F}}^m(x) G_{\mathcal{F}}^m(f)(x), & \text{if } b_\mu(x) \neq 0 \ \forall \mu = 1, \cdots, k+1 \\ P_{\mathcal{F}}^m(f)(x) & \text{if } x \in \text{some face } \mathcal{F} \end{cases} . \tag{55}$$

Here, the summation is over all facets $\mathcal{F}$ of the simplex $\mathcal{S}$. For the definition of the blending functions $c_{\mathcal{F}}^m(x)$ we let i be defined by $\mathcal{F} = \{x : b_i(x) = 0\}$

and let

$$c_{\mathcal{F}}^{m}(x) = \frac{\prod_{r \neq i} b_r^{\eta(m)}}{\sum_{j=1}^{k+1} \prod_{r \neq j} b_r^{\eta(m)}}. \tag{56}$$

where

$$\eta(m) = \begin{cases} m+2 & \text{if } m \text{ is even,} \\ m+1 & \text{if } m \text{ is odd.} \end{cases} \tag{57}$$

Thus the exponent in the blending functions is always even. This is required to maintain a convex combination, since some extrapolation may be necessary, leading to negative barycentric coordinates.

The operators $G_{\mathcal{F}}^{m}$ produce Taylor interpolants to derivatives perpendicular to the base face $\mathcal{F}$. They are defined by

$$G_{\mathcal{F}}^{m}(f)(x) = \sum_{\nu=0}^{m} \frac{b_i^{\nu}}{\nu!} P_{\mathcal{F}}^{m-\nu} \left(\frac{\partial^{\nu} f}{\partial s_{\mathcal{F}}^{\nu}} \right) \big(B_{\mathcal{F}}(x)\big) \tag{58}$$

where $\big(B_{\mathcal{F}}(x)\big)$ is the base point of the line of fixation with respect ot $\mathcal{F}$ through x, and $s_{\mathcal{F}} = V_i - B_{\mathcal{F}}(V_i)$. Note that $G_{\mathcal{F}}^{m}$ is defined in terms of lower dimensional perpendicular interpolants. If the operators $P_{\mathcal{F}}^{m-\nu}$ were replaced by the appropriate identity operator, then a transfinite scheme (interpolating everywhere on the boundary of $\mathcal{F}$) would emerge. To complete the recursion, we also define:

$$P_{\mathcal{S}}^{0}(f)\left(\sum_{i=1}^{k+1} b_i V_i\right) = \sum_{i=1}^{k+1} b_i f(V_i), \tag{59}$$

$$\dim(\mathcal{S}) = 0 \quad \Longrightarrow \quad P_{\mathcal{S}}^{m}(f)(x) = f(V) \quad (\text{since } \mathcal{S} = \{V\}) \tag{60}$$

and

$$\begin{aligned} \dim(\mathcal{S}) = 1 \quad \Longrightarrow \quad & P_{\mathcal{S}}^{m}(f)\big(tV_2 + (1-t)V_1\big) = \\ = \sum_{j=0}^{m} & \left[h_{0,j}^{m}(t) \frac{\partial^j f}{\partial(V_2 - V_1)}(V_1) + h_{1,j}^{m}(t) \frac{\partial^j f}{\partial(V_2 - V_1)}(V_2) \right], \end{aligned} \tag{61}$$

where the $h_{i,j}^{m}$ are cardinal polynomials of degree $2m+1$ defined uniquely by the properties

$$\left. \frac{\partial h_{i,s}^{m}(t)}{\partial x^{\mu}} \right|_{t\,=\,k} = \delta_{s\mu} \delta_{ik}, \tag{62}$$

$s, \mu \in \{0, 1, \cdots, m\}$, $i, k \in \{0, 1\}$.

See [7] for a more thorough discussion of perpendicular interpolants, including implementational details for $k \in \{2, 3\}$ and numerical examples.

8.3 A Transfinite Scheme

In [43], Gregory describes a transfinite simplicial scheme of arbitrary dimension k and arbitrary degree of smoothness m. The scheme is a convex combination of Boolean sums of Taylor type operators. It requires derivative data up to order m^k everywhere on the boundary of the simplex (and higher degree derivative data on lower dimensional faces). Its degree of precision is $k(m+1)-1$. A remarkable aspect of the scheme is that the Taylor operators yield interpolants that are polynomial along lines parallel to certain edges, and the blending functions can be chosen to be polynomial. Gregory also describes a rational alternative of the polynomial blending functions. (However, a transfinite scheme cannot be classified as rational or polynomial since its boundary data may render the interpolant non-rational.) Because of its generality, the scheme is a natural candidate for discretization, particularly for the case $m = 1$.

§9. Multivariate Splines

Perhaps the conceptually simplest approach to designing a multivariate interpolant is to generalize the concept of a *univariate spline*. Thus we assume that $\Omega \subset \mathbb{R}^k$ has been tessellated by subregions t_i, $i = 1, 2, \cdots, T$, and define

$$S_d^r(\Omega) = \left\{ s \in C^r(\Omega) \colon s|_{t_i} \in \mathbb{P}_d^k \quad \forall i = 1, 2, \cdots, k \right\} \tag{63}$$

The elements of $S_d^r(\Omega)$ are k-variate splines (of polynomial degree d and degree of smoothness r). We are of course particularly interested in the case that the subregions t_i form a k-dimensional triangulation. The hope in using $S_d^r(\Omega)$ instead of a finite element space, say, would be the ability to decrease the local complexity of a scheme (e.g., lower its polynomial degree), in exchange for the need to determine the parameters of the interpolant globally.

However, the global structure of multivariate splines turns out to be be extremely rich and complicated, and is at present only poorly understood. Even fundamental issues like the dimension and basis of spline spaces S_d^r have not been resolved in general, yet similar questions in the univariate case can be answered trivially. One of the major differences to the univariate case is that the dimension of S_d^r depends not just on the way in which simplices are connected, but also on the precise location of the data sites. Knowledge of *bivariate* splines is substantial and rapidly expanding. The subject was pioneered by Strang, [79], Morgan and Scott [54], Schumaker [67], [68]. The state of the art (as of this writing) is described in [14] and [46], and the references listed there.

A few results for trivariate splines are given in [8]. Following is a summary and an update:

Suppose for the moment that the 3-dimensional triangulation is shellable and has been built by adding one tetrahedron at a time, maintaining at each step a valid triangulation. Thus at each step we join the new tetrahedron to the developing triangulation at precisely 1,2, or 3 triangular facets. Let a_i,

$i = 1, 2, 3$ denote the *number of times* that we joined at precisely i facets. The values of a_i are crucial in expressing dimensions. They can be generalized to non-shellable triangulations by the definition

$$a_1 = N - 4, \quad a_3 = I, \quad \text{and} \quad a_2 = T - 1 - a_1 - a_3 \tag{64}$$

where T is the total number of tetrahedra and I is the number of interior vertices.

Because of the paucity of trivariate results we deviate briefly from our plan of considering only differentiable functions. It can be shown quite easily, using the Bézier-Bernstein form of a polynomial, that

$$\dim S_d^0 = \binom{\mathrm{d}+3}{3} + \binom{d+2}{3} a_1 + \binom{d+1}{3} a_2 + \binom{d}{3} a_3 \tag{65}$$

where

$$\binom{m}{n} = 0 \quad \text{if} \quad n > m \tag{66}$$

and $d \geq 1$. Let

$$P_d^r = \left\{ s \in S_d^0 : s \text{ is } r \text{ times differentiable at the vertices } x_i \right\}. \tag{67}$$

Then, for $d \geq 2r + 1$,

$$\dim P_d^r = \rho_0 + \sum_{i=1}^{3} \rho_i a_i \tag{68}$$

where

$$\rho_0 = \dim \mathbb{P}_d^3 = \binom{d+3}{3}, \tag{69}$$

$$\rho_1 = \binom{d+2}{3} - 3\binom{r+2}{3}, \tag{70}$$

$$\rho_2 = \binom{d+1}{3} - 2\binom{r+1}{3} - 2\binom{r+2}{3}, \tag{71}$$

and

$$\rho_3 = \binom{d}{3} - \binom{r}{3} - 3\binom{r+1}{3}. \tag{72}$$

For $d \geq 3$

$$\dim S_3^1 \geq \max \left\{ \rho_0, \rho_0 + \sum_{i=1}^{3} \tau_i a_i \right\} \tag{73}$$

where

$$\tau_1 = \big(d(d^2 - 1)\big)/6, \tag{74}$$

$$\tau_2 = \big(d(d^2 - 6d + 5)\big)/6, \tag{75}$$

and

$$\tau_3 = (d^3 - 12d^2 + 29d - 18)/6. \tag{76}$$

The formulas in [8] also contain a term accounting for *singular edges* that may be incorrect if several singular edges emanate from one point.

A similar formula is given in [8] for the case $r = 2$.

The lower bound (73) gives the exact dimension in all cases where the dimension has been calculated explicitly [9], at least for the generic case. Note, however, that the lower bound is given as the maximum of $\rho_0 = \dim \mathbb{P}_d^3$ and another term B, say. The reason is that for some 3-dimensional triangulations B may actually be less than ρ_0, whereas trivially $\dim S_d^1 \geq \dim \mathbb{P}_d^3 = \rho_0$. In [8] there is an example of a (large) triangulation where S becomes negative if $r \in \{3, 4\}$.

No useful *upper bounds* are known (excluding trivial ones like $\dim S_d^r \leq \dim P_d^r$). Other wide open questions include the existence of local bases, and the ability to solve the interpolation problem (2).

Specific trivariate examples where the dimensions changes with the geometry of the underlying triangulation are given in [9]. In addition, the author distributes periodically a *Multivariate Spline Newsletter* [10] that describes recent developments in the theory of multivariate splines, and lists further examples of geometric degeneracies.

The traditional approach to utilizing multivariate splines is to employ suitable subspaces of S_d^r. Finite elements constitute an extreme example where the polynomial degree is high, and the subspaces allow the construction of a minimally supported basis. A recently developed intermediate set of subspaces are those of *supersplines* and *vertex splines* [30]. These can be defined in any number of variables. A *superspline* is a spline $s \in S_d^r$ that is a specified number $R \geq r$ times differentiable at every vertex of the tessellation. For example, the finite element described in section 7.1, applied to every tetrahedron of a triangulation, defines a spline in $s \in S_d^1$. Since it is four times differentable at each vertex, it is also a superspline with $R = 4$. Moreover, since it has nonic precision, every superspline in S_9^1 with $R = 4$ can be defined locally by the finite element. A *κ-vertex spline* v (where $\kappa < k$) is a minimally supported superspline whose support is the star of a κ-face. Of particular interest for applications are spaces spanned by vertex splines. The finite element in section 7.1 can be used to construct κ-vertex splines for $\kappa = 0, 1, 2$. Chui and Lai [30] also introduce supersplines and vertex splines defined on a tessellation comprised of a mixture of simplices and parallelpipeds.

§10. Transfinite Hypercubal Methods

Occasionally there are applications in which transfinite schemes are appropriate. Examples include situations where a discretizing tool is moved across a surface, creating dense data along sparsely placed lines. We consider the case where the lines may be assumed to form a rectangular mesh. It is then reasonable to design schemes based on the assumption that functional data are given *everywhere* on the boundary of the tessellating rectangles. The natural generalization of a rectangle to higher dimension is a *hypercube* which can be scaled and represented by a cube $[0,1]^k$. For the case $k = 2$, several transfinite schemes on rectangles have been proposed. In [23], Barnhill and Worsey generalize this concept of a rectangular C^1 transfinite patch to an arbitrary number $k \geq 2$ variables. The difficulty in designing such schemes is to satisfy compatibility requirements on mixed partial derivatives. This is accomplished by incorporating suitable rational correction terms. The resulting schemes can be discretized if desired. In [83], Worsey describes a transfinite hypercubal C^2 scheme.

§11. Derivative Generation

It seems inevitable that in order to obtain an interpolant that is both local and smooth one has to supply derivative data. Typically, such data are not part of the interpolation problem and have to be made up from existing functional data. This process is usually referred as *derivative estimation*, but this is probably a misnomer. The objective is *not* to estimate existing but unknown values of derivatives. Instead, it is to generate values that will yield a *satisfactory interpolant*. Even if an underlying primitive function did exist it might be preferable to use derivative values that differ from the exact ones. (For example, a maximum error might be decreased by using the "wrong" derivative values.) Therefore, I prefer the term *Derivative Generation* rather than *Derivative Estimation*.

Most existing derivative generation schemes have been developed specifically for bivariate problems. However, most of the underlying ideas apply in more than two variables. It is desirabble to generate derivative values locally in order to keep the overall scheme local. On the other hand, numerical experiments [40] generation schemes tend to yield better results.

We now list a few specific local techniques:

1. Akima [1] discusses several versions of a gradient generation scheme that is based on averaging the normals defined by pairs of vectors connecting data points. Second derivatives can be obtained by applying the technique to gradient values.
2. Sibson [77] describes a specific scheme designed (for an arbitrary number of variables) in the spirit of his *natural neighbor* technique.
3. Stead [78] compares several techniques, including:
   - A Shepard based method where, for each data point, Shepard's interpolant is computed for the six closest sites (excluding the data site itself), and differentiated at the data site.
   - A similar Hardy Multiquadric based method where the interpolant is computed for twenty sites in a rectangular domain including the data site itself. That interpolant is then differentiated at the data site.
   - A weighted Least Squares approach where the gradients are obtained by differentiating a quadratic approximation to the data at the 10 closest sites plus the data site.

— Similarly, a linear squares fit of nine points.
— Little's method [8] which combines Shepard's method and triangulations.

On the basis of her experiments, Stead favors the multiquadric based generation technique.

All of the schemes described above have the significant advantage of being local. But they are also ad hoc schemes designed with a view to convenience and not towards desirable mathematical or physically meaningful properties. They may or may not perform well on specific problems, and choices must be based primarily on numerical experiments.

My own favorite approach is to select the interpolant from an affine space of many possible interpolants by requiring that a suitable functional be minimized. The *space* can be chosen conveniently to obtain interpolants with certain properties, such as being piecewise polynomial, easy to evaluate, or being of a structure that models behavior intrinsic in the physical problem (for example exponential decay). The *functional* can also be chosen to be meaningful in the context of the particular physical problem. For example, when building a ship by clamping metal plates to a frame work, one might minimize the clamped elastic plate functional (31) over a space of multivariate splines or finite elements. The resulting interpolant can be used as such, or required derivative data can be obtained by differentiating and evaluating the interpolant. These data might then be exploited in the construction of a different, perhaps smoother, interpolant. The advantages of this technique are its flexibility, and its underlying physical motivation. Its severe disadvantage is that in its unmodified form the derivative generation scheme is global rather than local. Some bivariate implementations and numerical examples are given in [5] and [6].

§12. Interpolation on the sphere and other surfaces

A special—but important and widely encountered—problem arises when the data sites lie on a 2-dimensional surface embedded in $\mathbb{R}^3$. Since we live on an approximately spherical planet, the most important instance of such a surface is a sphere. It is often unsatisfactory to project the surface, or different parts of it, into the plane. Instead, special methods have to be designed. Lawson [51], Renka [65], and Nielson and Ramaraj ([60] and [64]) independently propose schemes based on a triangulation of the surface of a sphere. In all of these schemes, the triangulations (by segments of great circles, rather than straight lines) are built by first constructing an initial triangulation and then swapping diagonals in convex quadrilaterals if required. Lawson's interpolant is the Clough-Tocher scheme [80]. His article also contains a detailed discussion of data structures, the triangulation procedure (yielding a generalization of the Delaunay triangulation), and a specific derivative generation scheme. Renka's scheme is a modification of his earlier planar scheme ([66] and [32]). His paper also contains algorithmic details. Nielson and Ramaraj use an interpolant based upon a minimal norm network (described for the planar case in [59]). Their paper contains several pictorial illustrations of their scheme. The surface can be rendered either as a shaded surface surrounding the sphere, or as a contour plot drawn on the sphere.

Wahba [81] describes an approach to *approximation* on the sphere that is based on *periodic* functions. Her technique can be specialized to interpolation. The paper also addresses the issues of noise in the data, smoothing splines, and cross-validation of results.

More general convex surfaces are considered by Barnhill *et al* ([20] and [21]) However, their main object of interest is the pressure on the wing of a particular aircraft. They propose a complex scheme, some of whose ingredients are: decomposition of the domain into two parts (the top and bottom of the wing), a multistage interpolation scheme yielding a bicubic map, Shepard's method, and Dirichlet tessellations.

Conclusions

The development of viable multivariate interpolation schemes is a rich and stimulating research area that is still in its infancy. The variety of problems requires a corresponding variety of techniques, schemes, and algorithms. Several techniques have been illustrated—or at least referenced—in this paper. These include, among others: tessellations of the domain; triangulations of the data sites; obtaining localness by expressing schemes solely in terms of data on a subdomain; using derivative data to increase the precision of a scheme, and making it local; generating derivative data from functional data; Boolean Sums; Convex Combinations; cardinal basis functions; blending partial schemes to obtain a complete one; the use of discontinuous ingredients to obtain smooth interpolants; transfinite schemes; discretization of transfinite schemes; the removal of compatibility conditions by adding rational correction terms; constructing high degree piecewise polynomial finite elements; constructing low degree finite elements on subdivided simplices; mollifying functions; interpolation to Taylor operators; recursion to obtain schemes on simplices of any dimension; judicious handling of perpendicular cross-boundary derivatives; radial functions; minimizing suitable linear functionals to select particular interpolants; the Bézier-Bernstein form of a (multivariate) polynomial. Naturally, there is a large number of sophisticated details to each of these techniques which could not be discussed in the given space.

We also encountered problems that do not arise in bivariate circumstances, for example: rendering difficulties, and the complexities of high-dimensional triangulations. The biggest difficulty in practical work has been the inability to adequately picture relevant phenomena. Whereas there is now a plethora of interpolation schemes for bivariate data, there exists only a handful of trivariate and higher-dimensional schemes, and in particular there is a paucity of algorithms.

One of the most significant phenomena in Multivariate Interpolation is that while there exist some highly sophisticated, complex and powerful schemes with attractive properties such as localness and arbitrary precision, yielding piecewise polynomial or rational interpolants, the schemes that are actually used by practitioners are mostly long known ad hoc schemes such as Shepard's method or Hardy Multiquadrics. Without offense to these marvelous pioneering schemes, it seems there should be some promise and incentive in exploring simplex based schemes, particularly when considering the increasing availability of parallel processing. The reason why this opportunity is not taken seems to be that the developers of complicated schemes, including this author, have failed to make the results of their work available as portable and well documented software. It is indeed unreasonable to expect a user—whose interests are in a field other than Mathematics—to spend the time and effort first to decipher the terse published description of a scheme and then program and debug it. On the other hand, the interests and abilities of the mathematicians who design their complicated schemes do not entice nor enable them to write the software themselves. What seems to be urgently required is the collaboration between a mathematician and a software engineer in producing widely usable software. This author would be interested in such collaboration.

Most schemes described here (and all of my own schemes) have been programmed as (often highly idiosyncratic and inefficient) research codes. The development of my own codes frequently uncovered subtle issues that had previously been overlooked, and sometimes outright errors and misconceptions. Indeed, I would be suspicious of any scheme that has not been actually tested on a computer. Indispensable in my work proved the FORTRAN software package MICROSCOPE [12]. This is an interactive program that enables the examination of functions (including multivariate interpolants) with respect to smoothness, precision, polynomial degree, and bugs in the implementation. The package is available (free of charge) via electronic mail from the *netlib* software distribution center [33] (and of course also from the author).

Another interesting phenomenon is that the best *tetrahedral* schemes (in my opinion, the piecewise cubic generalized Clough-Tocher split and the arbitrarily smooth and precise piecewise rational perpendicular interpolants) are actually special cases of more general interpolation schemes of arbitrary dimension. It would be reasonable to expect that better schemes could be designed by exploiting the special case $k = 3$. That this has not been the case illustrates the value of generalization and abstraction as a research tool.

Acknowledgments

I would like to thank the organizers of the Oslo conference for inviting me to present this survey at the conference, and to write this paper. The conference was very well run and a great pleasure to attend. My own work has benefited over many years from the fruitful and constructive interaction with many people. In particular, I would like to thank Hiroshi Akima, Bob Barnhill, Nelson Beebe, Luis Billera, Carl de Boor, Ron DeVore, Gerald Farin, Tom Foley, Dick Franke, Fred Fritsch, Tom Grandine, Klaus Hoellig, Tom Jensen, John Gregory, Chuck Lawson, Frank Little, Tom Lyche, Ed Nadeler, Greg Nielson, Bruce Piper, Paul Sablonnière, Kes Salkauskas, Ray Sarraga, Larry Schumaker, Frank Stenger, Keith Unsworth, and Andrew Worsey. In addition, my work has benefited greatly from the congenial environment and the excellent computing facilities at the University of Utah.

References

(Note: References not contained in the bibliography [41] in the form given here are marked with a superscript$^+$.)

1. Akima, H., On estimating partial derivatives for bivariate interpolation of scattered data, Rocky Mountain J. Math. **14** (1984), 41–52.
2. Alfeld, P., A discrete C^1 interpolant for tetrahedral data, Rocky Mountain J. Math. **14** (1984), 5–16.
3. Alfeld, P., A Trivariate Clough-Tocher Scheme for Tetrahedral Data, Comput. Aided Geom. Design **1** (1984), 169–181
4. Alfeld, P., A Trivariate Clough-Tocher Scheme for Tetrahedral Data, MRC Technical Summary Report #2702, Mathematics Research Center, University of Wisconsin–Madison, 1984.
5.$^+$ Alfeld, P., Triangular Extrapolation, MRC Technical Summary Report #2707, Mathematics Research Center, University of Wisconsin–Madison, 1984.
6. Alfeld, P., Derivative generation from multivariate scattered data by functional minimization, Comput. Aided Geom. Design **2** (1985), 281–296.
7. Alfeld, P., Multivariate Perpendicular Interpolation, SIAM J. Numer. Anal. **22** (1985), 95–106.
8.$^+$ Alfeld, P., On the dimension of multivariate piecewise polynomials, in D.F. Griffiths and G.A. Watson (ed.), *Numerical Analysis*, pp. 1–23, Longman Scientific & Technical, 1985.
9.$^+$ Alfeld, P., 1987, A Case Study of multivariate piecewise polynomials, in [36], pp. 149-160.
10.$^+$ Alfeld, P., The Multivariate Spline Newsletter, distributed periodically, to subscribe contact the author.
11. Alfeld, P., and R. E. Barnhill, A transfinite C^2 interpolant over triangles, Rocky Mountain J. Math. **14** (1984), 17–40.
12.$^+$ Alfeld, P. and B. Harris, Microscope: a smoothness tester for multivariate functions, MRC Rpt. 2701, Univ. Wisconsin-Madison, 1984.
13.$^+$ Alfeld, P., B. Piper, and L.L. Schumaker, Minimally Supported Bases for Spaces of Bivariate Piecewise Polynomials of Smoothness r and Degree $d \geq 4r+1$, Comput. Aided Geom. Design **4** (1987), 105–124.
14. Alfeld, P., B. Piper, and L. L. Schumaker, An explicit basis for C^1 quartic bivariate splines, SIAM J. Numer. Anal. **24** (1987), 891–911.
15. Barnhill, R. E., Representation and approximation of surfaces, in *Mathematical Software III* J. R. Rice (ed.), Academic Press, New York, 1977, 68–119.
16. Barnhill, R. E., Surfaces in computer aided geometric design: A survey with new results, Comput. Aided Geom. Design **2** (1985), 1–17.
17. Barnhill, R.E., R.P.Dube, and F.F. Little, Properties of Shepard's Surfaces, Rocky Mountain J. Math. **13**, 365–382.

18. Barnhill, R.E., and J.A. Gregory, Polynomial Interpolation to Boundary Data on Triangles, Math. Comp. **29** (1975), 726–735.
19. Barnhill, R.E., and F.F. Little, Three and Four-Dimensional Surfaces, Rocky Mountain J. Math. **14** (1984), 77–102.
20. Barnhill, R.E., B.R. Piper, and K.L. Rescorla, Interpolation to arbitrary data on a surface, in [36], 281–290
21.+ Barnhill, R.E., B.R. Piper, and S.E. Stead, A multidimensional surface problem: Pressure on a wing, Comput. Aided Geom. Design **2** (1985), 185–186.
22. Barnhill, R. E. and S. Stead, Multistage trivariate surfaces, Rocky Mountain J. Math. **14** (1984), 103–118.
23. Barnhill, R. E. and A. J. Worsey, Smooth interpolation over hypercubes, Comput. Aided Geom. Design **1** (1984), 101–113.
24.+ Bing, R.H., Some aspects of the topology of 3-manifolds related to the Poincaré conjecture, in *Lectures on Modern Mathematics*, v. 2, 93–128, Wiley, New-York, 1964.
25. de Boor, C., *A Practical Guide to Splines*, Springer Verlag, New York, 1978.
26. de Boor, C., Efficient computer manipulation of tensor products, ACM Trans. Math. Software **5** (1979), 173–182.
27. de Boor, C., B-form Basics, in [36], 131–148.
28.+ Buhmann, M.D., Convergence of Univariate Quasi-Interpolation using Multiquadrics, IMA J. Numer. Anal., to appear.
29.+ Buhmann, Multivariate Interpolation in Odd Dimensional Euclidean Spaces using Multiquadrics, submitted for publication.
30. Chui, C.K., and M. Lai, On Multivariate Vertex Splines and Applications, in [31], 19–36.
31. Chui, C.K., L.L. Schumaker, and F.I. Utreras, 1987, *Topics in Multivariate Approximation*, Proceedings of an international workshop, University of Chile, Santiago, Chile, December 15–19, 1986, Academic Press.
32. Cline, A.K., and R.J. Renka, A storage efficient method for construction of a Thiessen triangulation, Rocky Mountain J. Math. **14** (1984), 119–140.
33.+ Dongarra, J.J., and E. Grosse, Distribution of Mathematical Software Via Electronic Mail, Numerical Analysis Manuscript 85-2, AT&T Bell Laboratories, Murray Hill, New Jersey 07974.
34. Duchon, J., Interpolation des Fonctions de deux variables suivant le principle de la flexion de plaques minces, R.A.I.R.O. Analyse Numeriques **10** (1976), 5–12.
35.+ Dyn, D., 1987, Interpolation of scattered data by radial functions, in [31], 47–61.
36. Farin, G.E., (ed.) *Geometric Modeling*, SIAM Publication, 1987.
37. Farwig, R., 1986, Rate of Convergence of Shepard's Global Interpolation Formula, Math. Comp. **46**, 577–590.
38. Foley, T.A., Smooth Multivariate Interpolation to Scattered Data, Ph.D. thesis, 1979, Arizona State University.

39. Foley, T.A., Three-stage interpolation to scattered data, Rocky Mountain J. Math. **14** (1984), 141–150.
40. Franke, R.H., Scattered data interpolation: test of some methods, Math. Comp. **38** (1982), pp. 181-200.
41.+ Franke, R.H. and L.L. Schumaker, A Bibliography of Multivariate Approximation, in [31], 275–335
42. Gordon, W.J., and J.A. Wixom, 1978, On Shepard's method of "metric interpolation" to bivariate and multivariate interpolation, Math. Comp. **32** (1978), 253–264.
43. Gregory, J. A. Interpolation to boundary data on the simplex, Comput. Aided Geom. Design **2** (1985), 43–52.
44. Hardy, R. L., Multiquadric equations of topography and other irregular surfaces, J. Geophysical Res. **76** (1971), 1905–1915.
45.+ Hazelwood, C., A Divide and Conquer Approach to D-Dimensional Triangulations, Ph.D. Thesis, Department of Computer Science, University of Texas at Austin, 1988.
46.+ Ibrahim, A.K., and L.L. Schumaker, Super Spline Spaces of Smoothness r and Degree $d \geq 3r + 2$, preprint.
47.+ Jackson, I.R.H., Convergence Properties of Radial Basis Functions, Constr. Approx. **4** (1988), 243–264.
48.+ Kirkpatrick, J., private communication, 1988.
49. Lancaster, P., and K. Salkauskas, Surfaces generated by moving least squares methods, Math. Comp. **37** (1981), 141–158.
50.+ Lasser, D., Bernstein-Bézier-Darstellung trivariater Splines, Ph.D. Dissertation 1987, Technische Hochschule Darmstadt, FRG.
51. Lawson, C.L., C^1 surface interpolation for scattered data on the surface of a sphere, Rocky Mountain J. Math. **14** (1984), 177–202.
52. Lawson, C.L., Properties of n-dimensional triangulations, Comput. Aided Geom. Design **3**, (1986), 231-247.
53.+ Lawson, C.L., private communication, 1988.
54. Le Méhauté, A., Unisolvent Interpolation in $\mathbb{R}^n$ and the simplicial polynomial finite element method, in [28], 141–151.
55. Light, W. A. and E. W. Cheney, *Approximation Theory in Tensor Product Spaces*, Springer-Verlag Lecture Notes 1169, New York, 1985.
56. Little, F.F., Convex Combination Surfaces, in R.E. Barnhill and W. Boehm (eds), "Surfaces in Computer Aided Geometric Design", North-Holland, Amsterdam, pp. 99–108.
57. Micchelli, C.A., 1986, Interpolation of scattered data: distance matrices and conditionally positive definite functions, Constr. Approx. **2**, 11-22.
58. Morgan, J. and R. Scott, A nodal basis for C^1 piecewise polynomials in two variables, Math. Comp. **29** (1975), 736–740.
59. Nielson, G.M., A method for interpolating scattered data based upon a minimum norm network, Math. Comp. **40** (1983), 253–271.
60.+ Nielson, G.M., and R. Ramaraj, Interpolation over a sphere based upon a minimum norm network, Comput. Aided Geom. Design **4** (1987), 41–57.

61. Petersen, C.S., B.R. Piper and A.J. Worsey, Adaptive Contouring of a Trivariate Interpolant, in [36], 385–396.
62. Powell, M.J.D., Radial basis functions for multivariable interpolation, in J.C. Mason and M.G. Cox (ed.), *Algorithms for approximation*, Proc. Shrivenham Conf, Clarendon Press, Oxford, 1987.
63. Powell, M.J.D., and M.A. Sabin, Piecewise quadratic approximations on triangles ACM Trans. Math. Software **3** (1977), 316–325.
64. Ramaraj, R., *Interpolation and Display of Scattered Data over a Sphere*, M.S. Thesis, Computer Science Department, Arizona State University, Tempe, Arizona, 1986.
65. Renka, R.L., Interpolation of data on the surface of a sphere, ACM Trans. Math. Software **10** (1984), 417–436.
66. Renka, R.L., and A.K. Cline, A triangle based C^1 interpolation method, Rocky Mountain J. Math. **14** (1984), 223–238.
67.+ Rescorla, K.L., C^1 trivariate polynomial interpolation, Comput. Aided Geom. Design **4** (1987), 237–44.
68. Rippa, S., Interpolation and smoothing of scattered data by radial basis functions, M. Sc. Thesis, Tel Aviv University, Tel Aviv, Israel, 1984.
69.+ Rudin, M.E., A unshellable triangulation of a tetrahedron, Bull. Amer. Math. Soc. **64** (1958), 90–91.
70.+ Rushing, T.B., *Topological Embeddings*, Academic Press, 1973.
71. Schumaker, L.L., On the dimension of spaces of piecewise polynomials in two variables, in *Multivariate Approximation Theory*, W. Schempp and K. Zeller (eds.), Birkhäuser, Basel, 1979 396–412.
72. Schumaker, L.L., Bounds on the dimension of spaces of multivariate piecewise polynomials, Rocky Mountain J. Math. **14** (1984), 251–264.
73. Sederberg, T.W., Piecewise algebraic surface patches, Comput. Aided Geom. Design **2** (1985), 53–59.
74.+ Sewell, G., Plotting Contour Surfaces of a Function of Three Variables, ACM Trans. Math. Software **14** (1988), 33–41.
75.+ Sewell, G., Algorithm 657: Software for Plotting Contour Surfaces of a Function of Three Variables, ACM Trans. Math. Software **14** (1988), 33–41.
76. Shepard, D., 1968, A Two-Dimensional Interpolation Function for Irregularly Spaced Data, Proc. 23 Nat. Conf. ACM, 517–524.
77. Sibson, R., A brief description of natural neighbor interpolation, in sl Interpreting Multivariate Data, D.V. Barnett (ed.), Wiley, New York, 1981, 21–36.
78. Stead, S., Estimation of Gradients from Scattered Data, Rocky Mountain J. Math. **14** (1984), 265–280.
79. Strang, G., Piecewise polynomials and the finite element method, Bull. Amer. Math. Soc. **79** (1973), 1128–1137.
80. Strang, G, and J. Fix, *An analysis of the finite element method*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
81. Wahba, G., Surface fitting with scattered noisy data on Euclidean D-space and on the sphere, Rocky Mountain J. Math. **14** (1984), 281–299.

82. Wahba, G. and J. Wendelberger, Some new mathematical methods for variational objective analysis using splines and cross validation, Monthly Weather Review **108** (1980), 36–57.
83. Worsey, A. J., C^2 interpolation over hypercubes, CAGD **2** (1985), 107–115.
84. Worsey, A.J., and G.E. Farin, An n-dimensional Clough-Tocher interpolant, Constr. Approx. **3** (1987), 99–110.
85. Worsey, A.J., and B. Piper, A trivariate Powell-Sabin interpolant, Comput. Aided Geom. Design, to appear.
86. Ženíšek, A., Polynomial Approximation on Tetrahedrons in the Finite Element Method, J. Approx. Theory **7**, (1973), 334–351.

Peter Alfeld
Department of Mathematics
University of Utah
Salt Lake City, Utah 84112, U.S.A.
arpanet: ALFELD@SCIENCE.UTAH.EDU

Supported in part by NSF Grant DMS-8701121.