Polynomial interpolation in several variables

C. de Boor

Carl de Boor, University of Wisconsin-Madison

0. PREAMBLE

I want to thank the organizers for inviting me to this meeting as it gives me the opportunity to help celebrate Sam Conte who gave me my first academic job. More than that, he provided my children with many years of summer camp in the wilds of New Hampshire and Wisconsin, and at least two of them think that those summers in New Hampshire were essential for their growing up (and I tend to agree with them).

Now I realize that Sam does not yet know what I am talking about, so I will explain. I was young and impetuous when I came to Purdue, and ready to complain about everything, including the courses I had to teach and the books I had to use. Sam's textbook was not exempted from these gratuitous comments. But, instead of becoming miffed or angry, Sam merely invited me to work with him on a revision. Now, that may have ruined the book, as far as Sam is concerned, for it made it a much harder book. But it and a later edition have continued to sell enough copies to allow me the luxury of sending my children to summer camps in faraway places, and for that my children and I will forever be grateful.

1. INTRODUCTION

One of the things I changed rather drastically in that textbook was the treatment of polynomial interpolation. I was then (and still am) much impressed with the efficiency of the divided difference notion. It is a somewhat tricky Notion for the beginning student, and its treatment in the current edition is still not quite right. Perhaps we will get it right in the next one. In any case, polynomial interpolation occurs in the first real chapter of the book since polynomial interpolation is fundamental to much of numerical analysis.

It has therefore been something of a puzzle and disappointment to me that there is not a theory of *multivariate* polynomial interpolation as elegant and convincing and as basic as the univariate theory.

The trouble is easy to spot: Univariate polynomial interpolation starts with the observation that, for every set Θ of $k+1$ points, and for every function f defined (at least) on Θ, there is exactly one polynomial p of degree $\le k$ that matches f at Θ, that is, for which

$$p_{|\Theta} = f_{|\Theta}.$$

Thus, when someone walks in with $k+1$ points on the line, we immediately reach for Π_k := polynomials of degree $\le k$ as the space from which to choose the interpolant to given data at those points. But if our point set Θ is a subset of the plane or, more generally, of $\mathbb{R}^d$ for $d > 1$, then we do not know what to reach for. We would like to reach again for Π_k, but now it is not always possible to come up with a Π_k whose dimension

$$\dim \Pi_k(\mathbb{R}^d) = \binom{k+d}{d}$$

matches the cardinality of Θ; for example, with two points in the plane, we have too many points for Π_0 and too few points for Π_1. Further, even if $\#\Theta = \dim \Pi_k$, it may not be possible to interpolate from Π_k to every f on Θ. For example, if we take three points in the plane, then we can usually interpolate at those points with a linear polynomial, but if these three points happen to lie on a line, then our given f has to be linear before we can find an interpolant from Π_1, and even if we do, there are now many different interpolants.

Thus, the difficulty has been to come up, for given $\Theta \subset \mathbb{R}^d$, with a polynomial space P for which the pair $\langle \Theta, P\rangle$ is **correct** in the sense that any f defined (at least) on Θ matches exactly one $p \in P$ on Θ, that is, for which the restriction map

$$P \to \mathbb{R}^\Theta : p \mapsto p_{|\Theta}$$

is invertible. Generically, any polynomial space P with $\dim P = \#\Theta$ would do. The difficulty with multivariate polynomial interpolation has been that the point sets Θ one usually deals with are anything but generic. They may be regular meshes or lie on (other) simple algebraic surfaces.

This difficulty has been dealt with in the past in at least three ways.

2. STANDARD APPROACHES

Firstly, most past work has been spent deriving conditions on the set Θ for $\langle \Theta, \Pi_k \rangle$ to be correct. While much of this work is a variation on the eternal theme that a matrix is invertible if and only if its determinant is not zero, some of it is truly inspired. For example, Chung and Yao [1] (see also [2]) start with a sequence $a_1, a_2, \ldots, a_n \in \mathbb{R}^d$ such that $0, a_1, a_2, \ldots, a_n$ are in **general position**, which means that no (proper) hyperplane can contain more than d of these $n+1$ points. This implies that

$$\forall I \subseteq \{1, \ldots, n\} \text{with } \#I = d \quad \exists! x_I \quad s.t.\ \forall i \in I \quad 1 + a_i{*}x_I = 0$$
$$\forall i \notin I \quad 1 + a_i{*}x_I \neq 0$$

since it implies that any d of the a_i must be linearly independent, thus providing that unique point x_I, but also implies that, in addition to the d points $a_i, i \in I$, no point a_i with $i \notin I$ can lie in the hyperplane

$$\{x \in \mathbb{R}^d : 1 + x * x_I = 0\}.$$

This shows that the functions

$$\ell_I(x) := \prod_{i \notin I} \frac{1 + a_i{*}x}{1 + a_i{*}x_I}$$

are well-defined and, being products of $n-d$ linear factors, are elements of Π_{n-d}, and satisfy the conditions

$$\ell_I(x_J) = \delta_{IJ}.$$

Thus, for arbitrary f,

$$\sum_I \ell_I \, f(x_I)$$

interpolates to f on

$$\Theta := \{x_I : I \subset \{1, \ldots, n\},\ \#I = d\}.$$

Further, since

$$\#\Theta = \binom{n}{d} = \dim \Pi_{n-d},$$

this is the unique interpolant to f on Θ from Π_{n-d}.

Altogether, this is a most elegant generalization of the Lagrange form familiar from univariate polynomial interpolation. Its failing is simple: It is rarely of help in the common situation that one is given Θ.

Secondly, an entirely different effort, along the lines of the Newton form, was started by Gasca and Maeztu [3] some years ago. I follow these authors in describing the idea in the bivariate context. They start with a first instalment Θ_1 of data points all on a straight line, $l_1(x) = 0$ say. The interpolating polynomial p_1 for these is chosen as the unique interpolating polynomial of appropriate degree that is constant along lines perpendicular to the data line $l_1(x) = 0$. (Actually, Gasca and Maeztu permit greater freedom in the choice of p_1, but this will suffice to get the basic idea across.) A second instalment Θ_2 of data points, all on some straight line $l_2(x) = 0$, is dealt with by constructing the unique polynomial p_2, of appropriate degree and constant along lines perpendicular to the second line, that matches the modified data

$$\frac{f - p_1}{l_1}$$

at Θ_2. This ensures that the polynomial

$$p_1 + l_1 p_2$$

matches f at $\Theta_1 \cup \Theta_2$. A set of points on a third data line leads to the interpolant

$$p_1 + l_1 p_2 + l_1 l_2 p_3$$

in which p_3 matches the modified function

$$\frac{f - p_1 - l_1 p_2}{l_1 l_2}$$

at the data points on the third line; and so forth.

This scheme has the advantage of providing an interpolant in a form that is efficient for evaluation. Further, it is not that difficult to add repeated interpolation points to achieve osculatory (that is, Hermite) interpolation. On the other hand, there may be many ways of writing Θ as a disjoint union of sets on straight lines, and there is, offhand, no reason to prefer one over any of the others. Also, compared to other possibilities, the degree of the resulting interpolating polynomial may be much higher than is necessary.

Finally, the most intriguing method for me was one I learned from the thesis of Kergin [4], and which seems to have been inspired by Pierre Milman (see, for example, [5]). Here, one interpolates at the $k+1$ points in Θ by polynomials of degree $\le k$, exactly as in the univariate case. Of course, one must then deal with all the additional degrees of freedom available from $\Pi_k(\mathbb{R}^d)$ when $d > 1$. These are used in the Kergin scheme to make sure that various mean-value theorems hold for the interpolant Kf to given f, of the following kind:

Mean-Value Conditions. *For every subset* T *of* Θ*, and for every homogeneous polynomial* q *of degree* $j := \#\mathrm{T} - 1$*, there exists some point* τ *in the convex hull of* T *at which* $q(D)(f - Kf)(\tau) = 0$*.*

Here and below, $p(D)$ is the constant-coefficient differential operator $\sum_\alpha c(\alpha)D^\alpha$ obtained by "evaluating" the polynomial $p : x \mapsto \sum_\alpha c(\alpha)x^\alpha$ at $x = D$.

Kergin proves that there is exactly one linear projector K on $C^{(k)}(\mathbb{R}^d)$ into Π_k that satisfies all the Mean-Value Conditions. This makes it possible even to let some Θ coalesce and thereby obtain Hermite interpolation in the limit. For example, if all the points coalesce at some point z, then Kf is necessarily the Taylor expansion of f, at z, to terms of order k.

Kergin interpolation is particularly close to univariate polynomial interpolation in the sense that, when applied to any "plane wave" $f : x \mapsto g(\vartheta{*}x)$ (with g some univariate function), then

$$(Kf)(x) = (I_{\vartheta*\Theta}g)(\vartheta{*}x),$$

with $I_{\vartheta*\Theta}g$ the *uni*variate interpolant, at the points

$$\vartheta{*}\Theta := \{\vartheta{*}\theta : \theta \in \Theta\},$$

to g. I am particularly fond of Kergin interpolation since it led Micchelli [6] to the recurrence relations for simplex splines and so started the outpouring of work on multivariate B-splines of the last ten years. But, as a means of multivariate polynomial interpolation, its greatest drawback is the fact that the interpolant it provides has a much higher degree than may be required.

Of course, I am free to make all these negative comments about other people's efforts because I am about to describe a new effort, by my colleague Amos Ron and me, that avoids all the difficulties I complained about. I leave it to you and others to complain about the flaws in our approach.

3. NEW APPROACH

The approach centers on constructing a map

$$\Theta \mapsto \Pi_\Theta$$

that assigns to each finite point set $\Theta \in \mathbb{R}^d$ a polynomial space Π_Θ for which $\langle \Theta, \Pi_\Theta \rangle$ is correct.

Since almost any polynomial space P with $\dim P = \#\Theta$ gives a correct $\langle \Theta, P \rangle$, it would be good to have some guidelines. I give now a commented list of desired properties, based on the list Amos Ron put together when he talked about our scheme a year ago at the Texas A&M Approximation Theory meeting.

P1: well-defined , that is, $\langle \Theta, \Pi_\Theta \rangle$ should be correct, regardless of the choice of Θ.

P2: continuity (if possible), that is, small changes in Θ should not change Π_Θ by much. There are limits to this. For example, if $\Theta \subset \mathbb{R}^2$ consists of three points, then one would usually choose $\Pi_\Theta = \Pi_1$. But, as one of these points approaches some point between the two other points, this choice has to change in the limit, hence it cannot change continuously. As it turns out (see [7]), our scheme is continuous at every Θ for which $\Pi_k \subseteq \Pi_\Theta \subseteq \Pi_{k+1}$ for some k.

P3: coalescence $\Longrightarrow$ osculation (if possible), that is, as points coalesce, Lagrange interpolation should approach Hermite interpolation. This, of course, depends on just how the coalescence takes place. If, for example, a point spirals in on another, then we cannot hope for osculation. But if, for example, one point approaches another along a straight line, then we are entitled to obtain, in the limit, a match at that point also of the directional derivative in the direction of that line.

P4: translation-invariance, that is,

$$\forall(p \in \Pi_\Theta, a \in \mathbb{R}^d)\ \ p(a + \cdot) \in \Pi_\Theta.$$

This means that Π_Θ is independent of the choice of origin, and it implies that Π_Θ is ***D*-invariant**, that is, it is closed under differentiation.

P5: coordinate-system independence, that is, a linear change of variables $x \mapsto Ax$ (for some invertible matrix A) should affect Π_Θ in a reasonable way. Precisely,

$$\forall(\text{invertible } A)\ \ \Pi_{A\Theta} = \Pi_\Theta \circ A^T.$$

This implies that Π_Θ inherits any symmetries that Θ may have. It also implies (with a line or two of argument) that each $p \in \Pi_\Theta$ is constant along any lines orthogonal to the affine hull of Θ.

P6: scale-invariance, that is,

$$\forall(p \in \Pi_\Theta, \alpha \in \mathbb{R})\ \ p(\alpha \cdot) \in \Pi_\Theta.$$

This implies that Π_Θ is spanned by homogeneous polynomials. Note that P4 and P6 together are quite restrictive in the sense that the only spaces of smooth functions satisfying P4 and P6 are polynomial spaces.

P7: minimal degree, that is, the elements of Π_Θ should have as small a degree as is possible, since we would like the same property for the resulting interpolant. Here is the precise description:

$$\langle \Theta, P \rangle \text{ correct} \quad \Longrightarrow \quad \forall j\ \dim P \cap \Pi_j \le \dim \Pi_\Theta \cap \Pi_j.$$

Equivalently,

$$\deg I_\Theta p \le \deg p \quad \text{for every } p \in \Pi.$$

This implies, for example, that if $\langle \Theta, \Pi_k \rangle$ is correct, then $\Pi_\Theta = \Pi_k$. In other words, in the most heavily studied case, namely of Θ for which Π_k is an acceptable choice, our assignment would also be Π_k.

P8: monotonicity, that is,

$$\Theta \subset \Theta' \Longrightarrow \Pi_\Theta \subset \Pi_{\Theta'}.$$

This makes it possible to develop a Newton form for the interpolant. Also, in conjunction with P2, P7 and P9, this ties our scheme closely to standard choices.

P9: Cartesian product $\Longrightarrow$ tensor product, that is,

$$\Pi_{\Theta\times\Theta'} \ = \ \Pi_\Theta \otimes \Pi_{\Theta'}.$$

In this way, our assignment in the case of a rectangular grid coincides with the assignment standard for that case. In fact, by P8, it coincides with the standard assignment even when Θ is a **shadow** subset of a rectangular grid

$$\mathop{\times}_{i=1}^{d} \{xi(1),\ldots,xi(\gamma(i))\},$$

that is, $\Theta = \{\theta_\alpha : \alpha \in \Gamma\}$ for

$$\theta_\alpha := (x1(\alpha(1)),\ldots,xd(\alpha(d))),$$

with

$$\alpha \in C_\gamma := \{1,\ldots,\gamma(1)\} \times \cdots \times \{1,\ldots,\gamma(d)\}$$

and Γ an **order-closed** subset of C_γ, that is, $\alpha \in \Gamma$ and $\beta \le \alpha$ implies $\beta \in \Gamma$. Thus,

$$\Gamma = \bigcup_{\alpha\in\Gamma} C_\alpha.$$

Since, for any $\alpha \in \Gamma$, the subset

$$\Theta_\alpha := \{\theta_\beta : \beta \in C_\alpha\}$$

of Θ is a Cartesian product of sets from $\mathbb{R}$, our assignment for it is necessarily

$$\Pi_\alpha := \mathrm{span}\{()^\beta : \beta \le \alpha\},$$

by P9. By P8, each such Π_α must be contained in Π_Θ, hence

$$\mathrm{span}\{()^\beta : \beta \in \Gamma\} \subset \Pi_\Theta,$$

and, since

$$\dim \Pi_\Theta = \#\Theta = \#\Gamma,$$

Π_Θ must coincide with that span. Here and below,

$$()^\beta : x \mapsto x^\beta := x_1^{\beta(1)} \cdots x_d^{\beta(d)}$$

is a self-evident notation for the power map.

P10: constructible, that is, it should be possible to produce Π_Θ in finitely many arithmetic steps.

This list is detailed enough to determine Π_Θ uniquely in certain simple situations. For example, if $\#\Pi_\Theta = 1$, then necessarily $\Pi_\Theta = \Pi_0$ (by P7). If $\#\Theta = 2$, then, by P5 and P7, necessarily $\Pi_\Theta = \Pi_1(\mathrm{affine}(\Theta)) :=$ all linear polynomials that are constant in any direction perpendicular to the affine hull of Θ, that is, to the straight line containing Θ. If $\#\Theta = 3$, then $\Pi_\Theta = \Pi_k(\mathrm{affine}(\Theta))$, with $k := 3 - \dim \mathrm{affine}(\Theta)$. The case $\#\Theta = 4$ is the first one that is not clear-cut. In this case, we again have

$$\Pi_\Theta = \Pi_k(\mathrm{affine}(\Theta)), \qquad k := 4 - \dim \mathrm{affine}(\Theta),$$

but only for $k = 1, 3$. When affine(Θ) is a plane, we can use P4-P6 to normalize to the situation that $\Theta \subset \mathbb{R}^2$ and $\Theta = \{0, (1,0), (0,1), \theta\}$, with θ, offhand, arbitrary. Since Π_1 is the choice for the set $\{0, (1,0), (0,1)\}$, this means that $\Pi_\Theta = \Pi_1 + \mathrm{span}\{q\}$ for some homogeneous quadratic polynomial q. While P4-P6 impose further restrictions, it seems possible to construct a suitable map $\mathbb{R}^2 \to \Pi_2^0 : \theta \mapsto q$ (into *homogeneous* quadratic polynomials) in many ways so that the resulting $\Theta \mapsto \Pi_\Theta$ has all the properties P1-P10, except P8 perhaps. But neither Amos Ron nor I have so far been able to show that there is only one map $\Theta \mapsto \Pi_\Theta$ satisfying all conditions P1-P10. (Added remark (1992): On the other hand, it can be shown (see [8]) that

$$\Pi_\Theta = \cap_{p_{|\Theta}=0} \ker p_\uparrow(D)$$

with $p_\uparrow$ the **leading term** of the polynomial p, that is, the unique homogeneous polynomial for which $\deg(p - p_\uparrow) < \deg p$.)

Of course, we did not make up the above list and then set out to find the map $\Theta \mapsto \Pi_\Theta$. Rather, Amos Ron noticed that the pair $\langle \Theta, (\exp_\Theta)_\downarrow \rangle$ is always correct, and this motivated us to study the assignment

$$\Pi_\Theta := (\exp_\Theta)_\downarrow.$$

To explain,

$$H := \exp_\Theta := \mathrm{span}(e_\vartheta)_{\vartheta \in \Theta}$$

with

$$e_\vartheta : x \mapsto e^{\vartheta \cdot x}$$

the **exponential** with **frequency** ϑ. Further, for any space H of smooth functions,

$$H_\downarrow := \mathrm{span}\{f_\downarrow : f \in H\},$$

with $f_\downarrow$, the **least** of f, the first nontrivial term in the power series expansion

$$f = f^{(0)} + f^{(1)} + f^{(2)} + \ldots$$

for f, in which $f^{(j)}$ is the sum of all terms of (homogeneous) degree j, all j. Thus, $f_\downarrow$ is the homogeneous polynomial of largest degree for which

$$f = f_\downarrow + \text{higher order terms}.$$

It is not difficult to verify that this assignment satisfies P4-P6, P8-P9, and I will take up P10 in a moment. But it may not be clear why this has anything to do with interpolation.

4. REPRESENTATION OF POINT EVALUATION BY AN EXPONENTIAL

To make the connection, you need to be aware of the fact that the rule

$$\langle p, f \rangle \;:=\; p(D)f(0)$$

defines a pairing between polynomials p and smooth functions f and that e_ϑ represents the linear functional $p \mapsto p(\vartheta)$ with respect to this pairing, that is,

$$\langle p, e_\vartheta \rangle = p(\vartheta).$$

Further, $\langle \cdot, \cdot \rangle$ is an inner product on (real) polynomials, as can be seen from the fact that

$$\langle p, q \rangle = \sum_\alpha \frac{(D^\alpha p)(0)(D^\alpha q)(0)}{\alpha!}, \qquad p, q \in \Pi. \tag{1}$$

This suggests (as detailed in [7]) the construction of a basis for $H_\downarrow$ of the form $g_{1\downarrow}, \ldots, g_{n\downarrow}$ so that $\langle g_{i\downarrow}, g_j \rangle = 0$ if and only if $i \neq j$, with $g_1, g_2, \ldots, g_n$ a basis for H constructed from a basis $f_1, f_2, \ldots, f_n$ for H by a variant of the Gram-Schmidt process. Specifically, with suitable $g_1, g_2, \ldots, g_{j-1}$ already available (and spanning the same space as $f_1, f_2, \ldots, f_{j-1}$), one would compute

$$g_j := f_j - \sum_{i<j} g_i \frac{\langle g_{i\downarrow}, f_j \rangle}{\langle g_{i\downarrow}, g_i \rangle},$$

thereby ensuring that

$$\langle g_{i\downarrow}, g_j \rangle = 0, \qquad i < j, \tag{2}$$

while $g_j \neq 0$ (by the linear independence of the f_i), and therefore $\langle g_{j\downarrow}, g_j \rangle \neq 0$. The further modification

$$g_i \;\leftarrow\; g_i - g_j \frac{\langle g_{j\downarrow}, g_i \rangle}{\langle g_{j\downarrow}, g_j \rangle}$$

does not disturb the biorthogonality in Equation 2 already achieved, and it guarantees that

$$\langle g_{j\downarrow}, g_i \rangle = 0, \qquad i < j.$$

In this way, one obtains a basis $g_1, g_2, \ldots, g_n$ for H for which

$$\langle g_{j\downarrow}, g_i \rangle = 0 \quad \iff \quad i \neq j.$$

But this implies that the matrix $\big(\langle g_{i\downarrow}, g_j \rangle\big)$ is invertible, hence (since $g_1, g_2, \ldots, g_n$ and $f_1, f_2, \ldots, f_n$ are bases for the same space) the matrix $\big(\langle g_{i\downarrow}, f_j \rangle\big)$ is also invertible. If we start this calculation specifically with $f_j := e_{\vartheta_j}$ for all j, then this last matrix equals

$$\big(g_{i\downarrow}(\vartheta_j)\big),$$

and this proves that the pair $\langle\{\vartheta_1,\ldots,\vartheta_n\},H_\downarrow\rangle$ is correct. Further, for given f,

$$\sum_i g_{i\downarrow}\frac{\langle f,g_i\rangle}{\langle g_{i\downarrow},g_i\rangle}$$

is the unique interpolant to f from $H_\downarrow=\Pi_\Theta$, with

$$\langle f,g_i\rangle := \sum_j a_{ij}f(\vartheta_j)$$

in case $g_i =: \sum_j a_{ij}f_j$.

5. NUMERICS

Actual calculations depend a bit on just how one intends to represent this interpolant. While it is possible in principle to use a Newton form, it seems, as a first try, sufficient to write the interpolant in power form. One would want to shift this form, for example by the average of the ϑ_j, to avoid an obvious source of bad condition. For simplicity, I will ignore here this shift. Further, it seems advisable to use the **modified power form**

$$p=\sum_\alpha \frac{|\alpha|!}{\alpha!}()^\alpha\ \frac{D^\alpha p(0)}{|\alpha|!},$$

since its evaluation by the following "nested multiplication" (or "Horner's scheme") is immediate. (I have not been able to find this technique in the literature, but I have not looked for it very hard, either.) In this scheme, one sets

$$c(\alpha):=\frac{D^\alpha p(0)}{|\alpha|!},\qquad |\alpha|=\deg p,$$

and generates from this

$$c(\alpha):=\frac{D^\alpha p(0)}{|\alpha|!}+\sum_{i=1}^d x_i c(\alpha+\mathbf{i}_i),\qquad |\alpha|=k,$$

for $k=\deg p-1,\deg p-2,\ldots,0$, with $\mathbf{i}_i$ the ith unit vector. This works because one obtains

$$c(0)=\sum_{|\alpha|\le\deg p}\frac{D^\alpha p(0)}{|\alpha|!}\ n_\alpha x^\alpha,$$

with n_α the number of different increasing paths to α from the origin through points of $\mathbb{Z}_+^d$. This number is

$$n_\alpha = \binom{|\alpha|}{\alpha} = \frac{|\alpha|!}{\alpha!},$$

hence $c(0) = p(x)$.

Thus the goal of the calculation are the numbers

$$\frac{D^\alpha p(0)}{|\alpha|!}$$

for the interpolant p, and the calculations involve the scalar product

$$\langle p, q\rangle = \sum_\alpha \frac{\big(D^\alpha p\big)(0)\big(D^\alpha q\big)(0)}{\alpha!}$$

with $p \in \Pi$ and q "smooth". This implies that it is sufficient in the calculations to deal with any function g entirely in terms of the (first few entries in the) corresponding vector

$$Dg := (D^\alpha g(0))$$

(except for the function f to be interpolated, for which we know, offhand, nothing other than the vector $f_{|\Theta} = \big(f(\vartheta) : \vartheta \in \Theta\big)$). Note that $D(g_\downarrow)$ is obtained from Dg by direct truncation, hence also the needed computational step of obtaining $g_\downarrow$ from g can be carried out trivially in terms of the vector Dg.

While actual calculations require the imposition of some ordering on the points $\vartheta \in \Theta$ and the integer vectors $\alpha \in \mathbb{Z}_+^d$, it is more convenient, and less messy notationally, not to stress this computational requirement. Thus, for the time being, I let the ϑ in Θ and the $\alpha \in \mathbb{Z}_+^d$ index themselves. This means that our calculations start with the matrix

$$V := \Big(D^\alpha e_\vartheta(0) : \vartheta \in \Theta,\ \alpha \in \mathbb{Z}_+^d\Big)$$

whose rows are indexed by $\vartheta \in \Theta$ and whose columns are indexed by $\alpha \in \mathbb{Z}_+^d$. Since $p(D)e_\vartheta = p(\vartheta)$ for any polynomial p, the matrix V is the **Vandermonde** matrix for Θ, that is,

$$V = \Big(\vartheta^\alpha : \vartheta \in \Theta,\ \alpha \in \mathbb{Z}_+^d\Big).$$

This suggests the following slight detour, and this detour provides some insight into the special nature of our asssignment $\Theta \mapsto \Pi_\Theta$.

6. CONNECTION TO GAUSS ELIMINATION

Consider, for the moment, the possibility that we have not yet made up our minds from which polynomial subspace P to interpolate at Θ. We could then consider all possible choices for P by looking at the linear system

$$V? = f_{|\Theta}. \tag{3}$$

Any solution c with all but finitely many of its entries zero provides a polynomial, namely the polynomial $p := \sum_\alpha ()^\alpha c(\alpha)$, that agrees with f on Θ, and vice versa. We could now try to determine particularly "good" interpolants p. A possible criterion is that p have smallest possible degree. We could achieve this by ordering the columns of V by degree, that is, by $|\alpha|$, and then applying **elimination**, that is, Gauss elimination with partial pivoting, to V, in just the way it is taught in Linear Algebra courses. The result is a factorization

$$LW = V,$$

with L unit lower triangular, and W in row echelon form. This means that there is a sequence $\beta_1, \beta_2, \ldots, \beta_n$ that is strictly increasing, in the same total ordering of $\mathbb{Z}_+^d$ that was used to order the columns of V, and so that, for some ordering $\{\vartheta_1, \vartheta_2, \ldots, \vartheta_n\}$ of Θ and for all j, the entry $W(\vartheta_j, \beta_j)$ is the first nonzero entry in the row $W(\vartheta_j, :)$ of W. This makes the square matrix

$$U := \Big(W(\vartheta_i, \beta_j) : i, j = 1, \ldots, n\Big)$$

upper triangular and invertible, and so provides the particular interpolant $\sum_i ()^{\beta_i} a(i)$, whose coefficient vector

$$a := (LU)^{-1}(f(\vartheta_1), \ldots, f(\vartheta_n)) \tag{4}$$

is obtainable from the original data $f_{|\Theta}$ by permutation followed by forward- and backsubstitution.

There is no reason to believe that the resulting sequence $\beta_1, \beta_2, \ldots, \beta_n$ always consists of consecutive terms. It is exactly this fact that has prevented the development of a simple theory of multivariate polynomial interpolation. Rather, elimination has to face the numerical difficulty of deciding when all the pivots available for the current step in the current column are "practically zero", in which case the pivot search is extended to the entries in the next column (and in any row not yet used as pivot row). But this can also be viewed positively. Just as partial row pivoting has the "smallness" of the factors L and U as its goal, so the additional freedom of column pivoting allowed here provides further means of keeping the factors L and U "small". The smaller these factors, the better is the condition of the corresponding basis $\big(()^{\beta_j} : j = 1, \ldots, n\big)$ for the polynomial space P selected, when considered as a space of functions on Θ.

Surprisingly, the computational process for Π_Θ outlined earlier differs from this straightforward approach in only one detail: the entries of V are grouped by polynomial degree. In effect, V is viewed as the matrix

$$V := \Big(D^k f_\vartheta(0)\Big) = \Big(\vartheta^k : \vartheta \in \Theta,\, k = 0, 1, 2, \ldots\Big), \tag{5}$$

with *vector* entries

$$\vartheta^k := \big(\vartheta^\alpha : |\alpha| = k\big).$$

Note that

$$D^k g := \big(D^\alpha g(0) : |\alpha| = k\big)$$

represents the nontrivial part of $D(g_\downarrow)$ in case $g_\downarrow$ has degree k. Now we cannot expect elimination to zero out all entries in the pivot column below the pivot row. We can merely expect to make these entries *orthogonal* to the pivot element. The particular scalar product relevant here is

$$\langle D^k g, D^k q\rangle := \sum_{|\alpha|=k} \frac{D^\alpha g(0) D^\alpha q(0)}{\alpha!},$$

since, with this definition and in terms of the scalar product in Equation 1 for polynomials defined earlier,

$$\langle g_\downarrow, q\rangle = \langle g_\downarrow, q^{(k)}\rangle \;=\; \langle D^k g, D^k q\rangle$$

in case $k := \deg g_\downarrow$.

It is now easy to verify (see [9]) that the earlier Gram-Schmidt-like algorithm, applied to $f_j := e_{\vartheta_j}$, $j = 1, \ldots, n$, is Gauss elimination with *column* pivoting applied to the matrix in Equation 5. Once this is understood, it is also understood that Gauss elimination with *row* pivoting (that is, with possible reordering of the points in Θ) is just as effective. In fact, row pivoting provides the mechanism for choosing a "good" order in which to introduce the interpolation points into the calculations. Note that elimination with row pivoting necessarily leads to the same polynomial space, since Π_Θ does not depend on any particular ordering of the points in Θ and, with the ordering suggested by Gauss elimination with row pivoting, the two algorithms coincide.

This last remark is but one example of the importance of the theoretical underpinnings provided by [7], even though the calculations turn out to be nothing more than Gauss elimination (with a twist). For example, is it obvious from these calculations alone that $\Pi_\Theta \subset \Pi_{\Theta'}$ in case $\Theta \subset \Theta'$, or that, during Gauss elimination with partial pivoting, the next column has to contain a nontrivial pivot if the current column fails to contain one?

7. COMPUTATIONAL DETAILS

The calculations can be organized as follows. At the jth step, one looks for a pivot of the current order k among the rows not yet used as pivot rows. This means that one looks for $i \geq j$ that maximizes

$$\frac{\langle D^k g_i, D^k g_i \rangle}{\langle D^k f_i, D^k f_i \rangle}.$$

Here and below, g_i denotes the function obtained from $f_i := f_{\vartheta_i}$ by the elimination process as carried out so far; specifically,

$$g_i \perp g_{l\downarrow} \text{ for } l < j \leq i.$$

A row interchange (in all pertinent matrices) is made to bring the relatively largest pivot "element" into row j; k_j is set to the current k; and the appropriate multiple

$$\mathtt{LU}(i,j) := \frac{\langle D^k g_j, D^k g_i \rangle'}{\langle D^k g_j, D^k g_j \rangle'} \tag{6}$$

of row j is subtracted from row i for all $i > j$. Here, the scalar product

$$\langle D^k g, D^k f\rangle' := \langle D^k g, D^k f\rangle k! = \sum_{|\alpha|=k} \frac{|\alpha|!}{\alpha!} D^\alpha g(0) D^\alpha f(0) \tag{7}$$

is used instead of $\langle D^k g, D^k f\rangle$, since this makes the requisite weights integers (the multinomial coefficients), but it does not change the ratios in Equation 6.

It seems computationally efficient to compute the entire column $\texttt{LU}(:,j)$ by Equation 6, for later use, but set

$$\texttt{LU}(j,j) := \langle D^k g_j, D^k g_j\rangle'.$$

It may of course happen that

$$\frac{\langle D^k g_i, D^k g_i\rangle}{\langle D^k f_i, D^k f_i\rangle} < \texttt{tol}, \quad \forall\, i \geq j,$$

with `tol` some necessarily assigned tolerance. Then it is time to increase the order k by one and look again. As claimed earlier, there must now be some nonzero pivot available (though there is no guarantee that it will pass our tolerance test). Since we have no way of knowing *a priori* what the maximal degree in Π_Θ is going to be, it seems best to generate the columns of V as needed. Thus, at this stage, we must generate

$$V_k := (\vartheta^\alpha : \vartheta \in \Theta,\ |\alpha| = k),$$

for the new value of k. It seems most efficient to assume that, at this point, we still have in hand V_{k-1}, hence we can generate V_k by the appropriate multiplication of the entries of V_{k-1} by the components of the ϑ_r. The initial V_0 is the $n \times 1$-matrix $[1, 1, \ldots, 1]^T$. Further, having recorded the earlier elimination steps $1, \ldots, j-1$ in `LU`, we can compute the vectors $D^k g_i$ from V_k by forward substitution, that is, by applying L_j^{-1} from the left, with L_j the unit lower triangular matrix that agrees with `LU` in its first $j-1$ columns and below the diagonal, and has zeros otherwise.

In the end, we have available in our working array all the relevant entries of the vectors $Dg_i = (D^\alpha g_i(0) : \alpha)$; hence we can construct our interpolant p in the form $\sum_i g_{i\downarrow} c(i)$, with

$$c := D^{-1} U^{-1} L^{-1} (f(\vartheta_1), \ldots, f(\vartheta_n)),$$

and with L, U, and D the unit lower triangular, unit upper triangular, and diagonal matrix, respectively, whose nontrivial terms we stored in the array `LU`. Actually, since we have used the scalar product in Equation 7, $c(i)$ is too small by a factor $(\deg g_i)!$, hence just right for the modified power form discussed earlier.

The algorithmic details will appear in [9], but two (bivariate) examples follow.

8. INTERPOLATION AT THE VERTICES OF A REGULAR HEXAGON

Figure 1 shows (part of) the polynomial interpolant to the data $f(\vartheta_j) = (-1)^j$, with

$$\vartheta_j := (\cos(2\pi j/6), \sin(2\pi j/6)), \quad j = 1, \dots, 6.$$

For six generic points in the plane, one expects to interpolate from Π_2 since its dimension is 6. but these particular six points lie on the unit circle, that is, the quadratic polynomial

$$p_2 := 1 - ()^{2,0} - ()^{0,2}$$

vanishes on Θ, so Π_2 cannot be correct for this Θ. Since any five of these points are linearly independent over Π_2, we know that Π_Θ has the form

$$\Pi_\Theta = \Pi_1 + \big(\Pi_2^0 \ominus \operatorname{span}(p_2)\big) + \operatorname{span}(p_3),$$

with the orthogonal complement in the space Π_2^0 of homogeneous second-degree polynomials taken in terms of the scalar product in Equation 1, and with p_3 a particular homogeneous cubic polynomial. In fact, p_3 coincides, up to a scalar multiple, with the interpolant depicted in Figure 1, for the following reason: By symmetry, there are three straight lines through the origin that do not contain any interpolation point and are such that reflection across any one of them leaves Θ invariant but changes the given function values to their negatives, hence this reflection must change the interpolant to its negative, and therefore the interpolant must vanish along these three lines. But this implies that all its derivatives of order ≤ 2 at the origin must be zero. This argument,

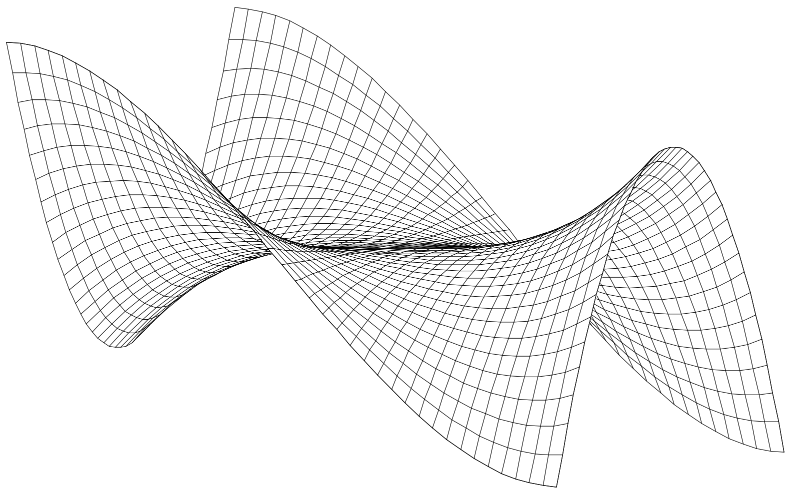

Figure 1. The cubic term in interpolation at the hexagon points

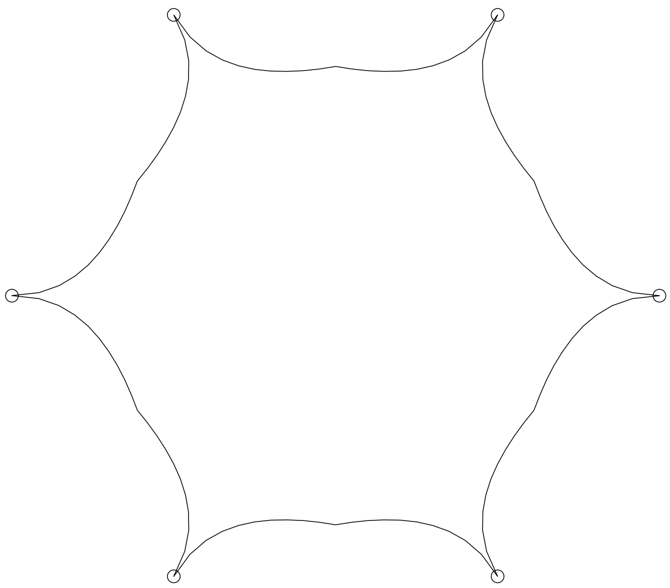

Figure 2. The Lebesgue function for interpolation at the hexagon points $\circ$ equals 1 on this entire domain.

incidentally, shows that, for this (highly symmetric) Θ, the Properties P1 - P10 uniquely determine Π_Θ.

This resolves in a simple way the following puzzle: Since in this case $A\Theta = \Theta$ for A := rotation by $\pi/3$, we know from Property P5 that $\Pi_\Theta = \Pi_\Theta \circ A$. Since, up to scalar multiples, p_3 is the unique cubic homogeneous polynomial in Π_Θ, this leads to the (careless) conclusion that p_3 must have the symmetry $p_3 = p_3 \circ A$. But that implies that p_3 is constant on Θ, hence necessarily coincident with the appropriate multiple of the constant function $()^0$ (note that Π_0 is contained in any Π_Θ, by Properties P7 and P8). The picture reminds us of the fact that, strictly speaking, we only know that $\mathrm{span}(p_3) = \mathrm{span}(p_3) \circ A$, for, according to the picture, $p_3 \circ A = -p_3$.

The Lagrange polynomial associated with the point (1,0) is given by

$$\ell(x) := ((1 + 2x_1 + 2x_1^2 + x_1^3) - x_2^2(2 + 3x_1))/6.$$

This makes it easy to determine its zero set, hence to see that it and its five rotates are nonnegative on a rather large portion of the hexagon. This domain is shown in Figure 2. At any point of this domain, the value of the interpolating polynomial is an average of the given function values. Equivalently, the Lebesgue function of the process (that is, the sum of the absolute values of all the Lagrange polynomials) is 1 on this entire domain. In univariate polynomial interpolation, the Lebesgue function is 1 on a set larger than just the interpolation points only for linear interpolation.

For the hexagon points, Π_Θ does not contain Π_2; hence the interpolant provides only a second-order approximation (as we let the diameter of the circle of points shrink). By also adding the center of the circle, Π_Θ becomes $\Pi_2 + \mathrm{span}(p_3)$ (by P7 and P8). The additional function is the polynomial p_2 mentioned earlier; it serves as the Lagrange polynomial for the new point. The other Lagrange polynomials are $\ell - p_2/6$ and its five rotates. Now the Lebesgue function equals 1 only at the interpolation points. But, as Figure 3 shows, the Lebesgue function does not exceed 1.5 on the hexagon spanned by the points, and it does not exceed 1.7 on the unit disk. This says that, as a map on continuous functions on the unit disk in the max-norm, this interpolation scheme has norm less than 1.7. That is remarkable.

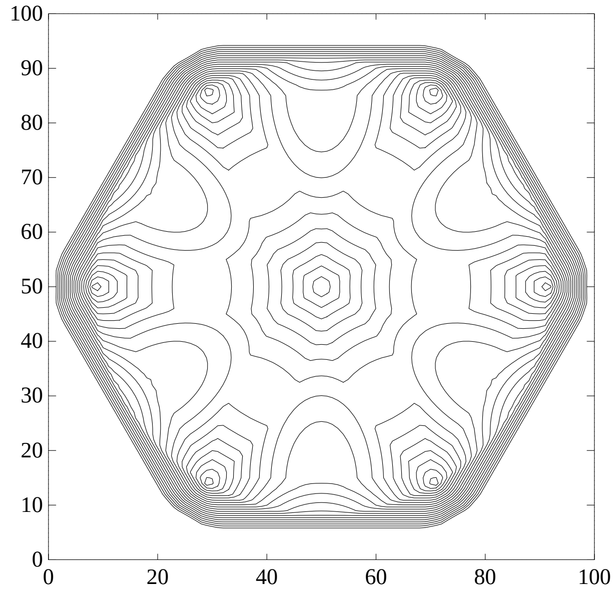

Figure 3. Contour lines, for values 1, 1.05, ..., 2, of Lebesgue function for interpolation at hexagon points and their center.

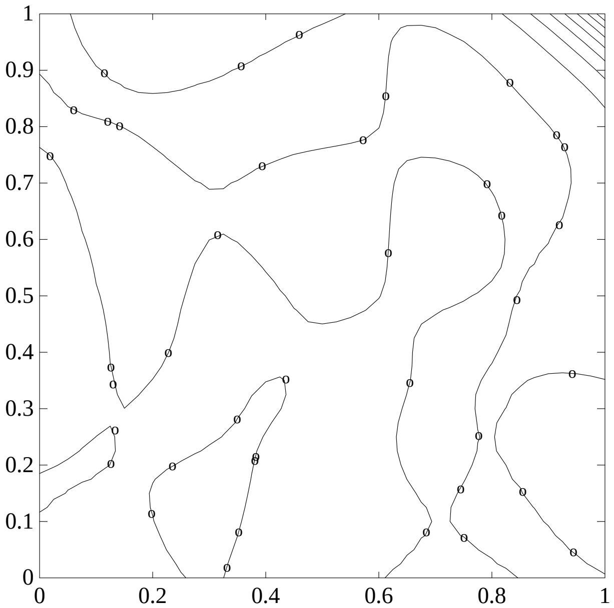

Figure 4. Contour lines for error in interpolation at 40 random points.

9. INTERPOLATION AT 40 RANDOMLY CHOSEN POINTS

Figure 4 shows contour lines (corresponding to ten equally spaced function values between maximum and minimum value) of the absolute error in the polynomial interpolant to

$$f : x \mapsto \exp(-x_1^2 - x_2^2)$$

at 40 points chosen at random from the square $[0 \,..\, 1]^2$. These interpolation points are also marked in Figure 4. Not surprisingly, they all fall on the zero contour line and so indicate that the error is near zero in most of the square. Only near the corners of the square is the error not close to zero. In fact, the maximum error on the square $[0 \,..\, 1]^2$ turned out to be `2.6e-4`. (The calculations were done with MATLAB, hence in roughly 16-decimal-digit arithmetic. The maximum difference between the input function values and the corresponding values of the computed interpolating polynomial was `7.8e-16`.) Examples like these are making me re-examine the standard conviction that polynomial interpolation at many points is not expected to be useful. It may well be that this is less true in several variables than in one, since, in several variables, the polynomial degree usually grows much slower than the number of data points if an interpolating polynomial of smallest possible degree is used. On the other hand, there is no reason to expect that the Lebesgue function behaves any better in several variables than in one.

10. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under grant no. DMS-8701275 and by the US Army under contract no. DAAL03-87-K-0030.

11. REFERENCES

[1] K. C. Chung and T. H. Yao (1977), "On lattices admitting unique Lagrange interpolations", *SIAM J. Numer. Anal.* **14**, 735–741.

[2] W. Dahmen and C. A. Micchelli (1981), "On limits of multivariate B-splines", *J. Analyse Math.* **39**, 256–278.

[3] M. Gasca and J. I. Maeztu (1982), "On Lagrange and Hermite interpolation in $\mathbb{R}^k$", *Numer. Math.* **39**, 1–14.

[3] Kergin, P. *Interpolation of C^k Functions*, Ph.D. diss., Univ. of Toronto, Canada, (1978); published as: "A natural interpolation of C^k functions." *J. Approx. Theory* **29**, 278–293 (1980).

[5] C. A. Micchelli and P. Milman (1980), "A formula for Kergin interpolation in $\mathbb{R}^k$", *J. Approx. Theory* **29**, 294–296.

[6] C. A. Micchelli (1980), "A constructive approach to Kergin interpolation in $\mathbb{R}^k$: multivariate B-splines and Lagrange interpolation", *Rocky Mountain J. Math.* **10**, 485–497.

[7] C. de Boor and A. Ron (1990), "On multivariate polynomial interpolation", *Constr. Approx.* **6**, 287–302.

[8] C. de Boor and A. Ron (1992), "The least solution for the polynomial interpolation problem", *Math. Z.* **210**, 347–378.

[9] C. de Boor and A. Ron (1992), "Computational aspects of polynomial interpolation in several variables", *Math. Comp.* **58**, 705–727.