UNIVERSITY OF WISCONSIN-MADISON
COMPUTER SCIENCES DEPARTMENT

Computational aspects of polynomial interpolation in several variables

Carl de Boor[1,2] & Amos Ron[1]

March 1990

ABSTRACT

The pair $\langle\Theta, P\rangle$ of a pointset $\Theta \subset \mathbb{R}^d$ and a polynomial space P on $\mathbb{R}^d$ is **correct** if the restriction map $P \to \mathbb{R}^\Theta : p \mapsto p_{|\Theta}$ is invertible, i.e., if there is, for any f defined on Θ, a unique $p \in P$ which matches f on Θ.

We discuss here a particular assignment $\Theta \mapsto \Pi_\Theta$, introduced in [3], for which $\langle\Theta, \Pi_\Theta\rangle$ is always correct, and provide an algorithm for the contruction of a basis for Π_Θ which is related to Gauss elimination applied to the Vandermonde matrix $(\vartheta^\alpha)_{\vartheta\in\Theta,\alpha\in\mathbb{Z}_+^d}$ for Θ. We also discuss some attractive properties of the above assignment and algorithmic details of the algorithm, and present some bivariate examples.

AMS (MOS) Subject Classifications: primary 41A05, 41A10, 41A63, 65D05, 65D15 secondary 15A12

Key Words: exponentials, polynomials, multivariate, interpolation, multivariate Vandermonde, Gauss elimination

Authors' affiliation and address:
Computer Sciences Department
University of Wisconsin-Madison
1210 West Dayton St.
Madison WI 53706

[1] supported by the United States Army under Contract No. DAAL03-87-K-0030
[2] supported by the National Science Foundation under Grant No. DMS-8701275

Computational aspects of polynomial interpolation in several variables

Carl de Boor & Amos Ron

We say that the pair $\langle\Theta, P\rangle$ of a (finite) pointset $\Theta \subset \mathbb{R}^d$ and a (polynomial) space P of functions on $\mathbb{R}^d$ is **correct** if the restriction map

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

is invertible, i.e., if there is, for any f defined (at least) on Θ, exactly one $p \in P$ which matches f on Θ, i.e., satisfies $p(\vartheta) = f(\vartheta)$ for all $\vartheta \in \Theta$.

Polynomial interpolation in one variable is so basic a Numerical Analysis tool that many textbooks on Numerical Analysis begin with this topic and none fails to provide a detailed account of it. The topic is associated with the illustrious names of Newton, Cauchy, Lagrange, Hermite, and is essential for various basic tasks, such as the construction of rules for quadrature and differentiation, or the construction of difference approximations for ordinary differential equations. To be sure, polynomial interpolation is not a general purpose tool for approximation, for the simple reason that polynomials are only good for *local* approximation (though for some particularly well-behaved function, this might mean approximation on the entire line). Even in local approximation, badly handled polynomial interpolation, such as interpolation at equally spaced points, is not to be recommended in general. But well-handled polynomial interpolation, such as interpolation at the Chebyshev points, is one of the most efficient ways available for local approximation.

It has been therefore all the more annoying that there has not been available a correspondingly simple and effective theory of multivariable polynomial interpolation. The reason is easy to spot: Whereas there is a unique interpolant from the space Π_k of polynomials of degree $\le k$ for any data given on any $k+1$-pointset in $\mathbb{R}$, there is no corresponding universal multivariable space of polynomials. In other words, a correct polynomial space P for interpolation to an arbitrary f at the given set $\Theta \subset \mathbb{R}^d$ cannot in general be determined from the cardinality $\#\Theta$ of the pointset Θ alone. Rather, the actual location and configuration of Θ must be taken into account. Further, the standard choice of $P = \Pi_k$ requires that $\#\Theta$ equal

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

Finally, even if Θ satisfies this rather restrictive requirement, there is no guarantee that the pair $\langle\Theta, \Pi_k\rangle$ is correct.

In [3], we give a particular assignment

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

for which $\langle\Theta, \Pi_\Theta\rangle$ is always correct, and give an algorithm for the construction of a basis for Π_Θ from Θ. We also prove there some of its nice properties. In the present paper, we list these and other properties of our assignment $\Theta \mapsto \Pi_\Theta$ and, eventually, verify the additional ones. We also provide some enticing (so we hope) examples. But the main point of the present paper is a detailed discussion of the algorithmic aspects of our particular choice: How is Π_Θ to be constructed and, once in hand, how is the interpolant from it to be found?

We did provide in [3] an algorithm for the construction of Π_Θ, but found to our surprise (cf. [2]) that Π_Θ can also be constructed by Gauss elimination applied to the Vandermonde matrix $\left(\vartheta^\alpha\right)$ for Θ, but with a twist. This allows us to view our particular assignment Π_Θ in retrospect as arising from a stabilization and symmetrization of a simple-minded approach for finding a correct polynomial space of minimal degree for interpolation at Θ.

The paper is organized as follows:

In Section 1, we recall necessary details from [3] concerning the definition of our polynomial interpolant, give a very simple verification of our formula for the interpolant, and give an extensive list of its properties.

In Section 2, we use Gauss elimination to extract from the Vandermonde matrix (ϑ^α) (for the given $\vartheta \in \Theta$) a monomial-spanned polynomial space of lowest possible degree which is correct for interpolation at Θ, and prove that the same calculation also provides a basis for Π_Θ, albeit not a very convenient one.

In Section 3, we show that Gauss elimination applied to the Vandermonde matrix, but carried out degree by degree rather than monomial by monomial, leads to a convenient basis for Π_Θ and provides a suitable ordering of the points of Θ, and contrast this with the algorithm proposed in [3] which corresponds to Gauss elimination by columns, with column pivoting without interchanges, and fails to provide an ordering of the points in Θ. Since, in our multivariable setting, each degree (other than degree 0) involves several monomials, we have to replace the standard goal of Gauss elimination, viz. the generation of zeros below the pivot element, by the more suitable goal of making the entries below the pivot element orthogonal to the pivot element, with respect to a certain weighted scalar product. We believe that such a generalization of Gauss elimination may be advantageous in other situations where more than partial pivoting is needed but total pivoting is perhaps too radical a measure.

In Section 4, we introduce a modified power form for multivariate polynomials as well as a nested multiplication algorithm for its efficient evaluation. We believe both the form and the algorithm to be new (with the algorithm closely related to de Casteljau's algorithm for the evaluation of the Bernstein-Bézier form).

In Section 5, we give a detailed description (in a MATLAB-like program) of the calculation of the modified power coefficients of our interpolant from the given data $(\vartheta, f(\vartheta))$, $\vartheta \in \Theta$.

We illustrate the interpolation procedure with three examples in Section 6: The first explores the first nontrivial case, that of a four-point set Θ coplanar but not collinear, the second illustrates the close connection of Π_Θ to polynomials which vanish on Θ, and the third shows that the algorithm works sufficiently well to provide the polynomial interpolant to a smooth bivariate function at 40 randomly chosen points. The second example also shows the surprising fact that our interpolant to data at the six vertices of a regular hexagon takes a *convex* combination of the given function values as its value at every point in a hexagon-shaped region, and makes the point that, for any Θ on some circle in the plane, our polynomial space Π_Θ consists of harmonic polynomials.

In Section 7, we provide discussion and proofs of the various properties listed in Section 1, and close with a short section on a generalization of our process, from point evaluations to arbitrary linear functionals on Π.

For alternative approaches to multivariable polynomial interpolation in the literature, see, e.g., their discussion in [2].

1. The interpolant and some of its properties

The **leading** term $p_\uparrow$ of a polynomial p is, by definition, the homogeneous polynomial for which $\deg(p-p_\uparrow) < \deg p$. The construction proposed in [3] makes use of an analogous concept for power series, namely the **initial** term $f_\downarrow$ of a function f analytic at the origin. This is the homogeneous polynomial $f_\downarrow$ for which $f - f_\downarrow$ vanishes to highest possible order at the origin. In other words, $f_\downarrow$ (we call it 'f least' for short) is the first nontrivial term in the power series expansion

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

for f in which $f^{(j)}$ is the sum of all the (homogeneous) terms of degree j.

For example, with $\vartheta \cdot x := \sum_{j=1}^{d} \vartheta(j)x(j)$ the ordinary scalar product of the two d-vectors ϑ and x, the **exponential** e_ϑ **with frequency** ϑ has the power series expansion

$$e_\vartheta(x) := e^{\vartheta\cdot x} = 1 + \vartheta\cdot x + (\vartheta\cdot x)^2/2 + \ldots .$$

Therefore,

$$(e_\vartheta)_\downarrow(x) = 1, \qquad (e_\vartheta - e_{\vartheta'})_\downarrow(x) = (\vartheta - \vartheta')\cdot x,$$

the latter in case $\vartheta \ne \vartheta'$.

We also use the abbreviation

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

for any linear space H of functions analytic at the origin, and recall from [3] the fact that

$$\dim H_\downarrow = \dim H. \tag{1.1}$$

In these terms, our assignment for Π_Θ is

$$\Pi_\Theta := (\exp_\Theta)_\downarrow, \tag{1.2}$$

with

$$\exp_\Theta := \operatorname{span}\{e_\vartheta : \vartheta \in \Theta\}.$$

Thus, if Θ consists of a single point, then $\Pi_\Theta = \Pi_0$, while if Θ consists of the two points ϑ, ϑ', then Π_Θ is spanned by the two polynomials 1, $(\vartheta - \vartheta')\cdot$, i.e., Π_Θ consists of the (two-dimensional) space of all polynomials which are (at most) linear in the direction $\vartheta - \vartheta'$ and constant in any direction orthogonal to $\vartheta - \vartheta'$.

The construction of our interpolant also makes use of the **pairing**

$$\langle g, f\rangle := \sum_\alpha D^\alpha g(0)\ D^\alpha f(0)/\alpha! \tag{1.3}$$

defined, e.g., for an arbitrary function g analytic at the origin and an arbitrary polynomial f. The weights in (1.3) are chosen so that point-evaluation at ϑ is represented with respect to this pairing by the exponential e_ϑ, i.e.,

$$\langle e_\vartheta, f\rangle = f(\vartheta), \qquad \vartheta \in \mathbb{R}^s,\ f \in \Pi, \tag{1.4}$$

as one readily verifies by substituting $\vartheta^\alpha = (D^\alpha e_\vartheta)(0)$ for $(D^\alpha g)(0)$ in (1.3). This justifies the following extension of the pairing to arbitrary $g \in \exp_\Theta$ and $f \in C(\mathbb{R}^d)$ by

$$\langle \sum_{\vartheta\in\Theta} w(\vartheta) e_\vartheta, f\rangle := \sum_{\vartheta\in\Theta} w(\vartheta) f(\vartheta), \qquad f \in C.$$

This extension is well defined since any collection of exponentials with distinct frequencies is linearly independent (see (2.4)Fact below). Consequently, $\dim \exp_\Theta = \#\Theta$, and

$$\sum_i w(i) g_i = e_\vartheta \quad \Longrightarrow \quad \sum_i w(i)\langle g_i, f\rangle = f(\vartheta)\ \text{for} f \in C. \tag{1.5}$$

The construction proposed in [3] provides the polynomial interpolant $I_\Theta f$ in the form

$$I_\Theta f := \sum_{j=1}^n g_{j\downarrow} \frac{\langle g_j, f\rangle}{\langle g_j, g_{j\downarrow}\rangle}, \tag{1.6}$$

with $g_1, g_2, \ldots, g_n$ a(ny) basis for $\exp_\Theta$ (hence, in particular, $n = \#\Theta$) for which

$$\langle g_i, g_{j\downarrow}\rangle = 0 \quad \Longleftrightarrow \quad i \neq j. \tag{1.7}$$

Since each $g_{j\downarrow}$ is a (homogeneous) polynomial, it is clear that $I_\Theta f$ is a polynomial. But it may be less obvious why $I_\Theta f = f$ on Θ. Here is a simple argument.

From (1.7), it follows that $I_\Theta f$ is well-defined and that

$$\langle g_i, I_\Theta f\rangle = \langle g_i, f\rangle, \quad \text{all } i. \tag{1.8}$$

Since $g_1, g_2, \ldots, g_n$ is a basis for $\exp_\Theta$, this implies (with (1.4) and (1.5)) that

$$I_\Theta f(\vartheta) = \langle e_\vartheta, I_\Theta f\rangle = \langle e_\vartheta, f\rangle = f(\vartheta), \quad \vartheta \in \Theta.$$

This also implies that the space

$$\operatorname{span}\{g_{1\downarrow}, \ldots, g_{n\downarrow}\}$$

is a correct polynomial space for interpolation at Θ. This space is contained in Π_Θ. But since $\dim \Pi_\Theta = \dim \exp_\Theta = \#\Theta = n$ (the first equality by (1.1)), we must have

$$\Pi_\Theta = \operatorname{span}\{g_{1\downarrow}, \ldots, g_{n\downarrow}\}.$$

In order to provide encouragement, we now list some nice properties of this particular map $\Theta \mapsto \Pi_\Theta$, but postpone their verification until after the discussion of the algorithm for the construction of the interpolant.

(1) well-defined, i.e., for any finite Θ, Π_Θ is a well-defined polynomial space and $\langle \Theta, \Pi_\Theta \rangle$ is correct.

(2) continuity (if possible), i.e., small changes in Θ shouldn't 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 $\Pi_\Theta = \Pi_1$ (as our scheme does). 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, our scheme is continuous at every Θ for which $\Pi_k \subseteq \Pi_\Theta \subseteq \Pi_{k+1}$ for some k.

(3) coalescence $\Longrightarrow$ osculation (if possible), i.e., as points coalesce, Lagrange interpolation approaches Hermite interpolation. This will, of course, depend on just how the coalescence takes place. If, e.g., a point spirals in on another, then we cannot hope for osculation. But if, e.g., 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.

(4) translation-invariance, i.e., $\forall (p \in \Pi_\Theta, a \in \mathbb{R}^d)\ \ p(a + \cdot) \in \Pi_\Theta$. This implies that Π_Θ is D-invariant, i.e., is closed under differentiation.

(5) scale-invariance, i.e., $\forall (p \in \Pi_\Theta, \alpha \in \mathbb{R})\ \ p(\alpha \cdot) \in \Pi_\Theta$. This is equivalent to the fact that Π_Θ is spanned by homogeneous polynomials. Note that **(4)** and **(5)** together are quite restrictive in the sense that the only finite-dimensional spaces of smooth functions satisfying **(4)** and **(5)** are polynomial spaces.

(6) coordinate-system independence, i.e., an affine change of variables $\vartheta \mapsto A\vartheta + c$ (for some invertible matrix A) affects Π_Θ in a reasonable way. Precisely,

$$\forall \{\text{invertible } \ A \in \mathbb{R}^{d\times d}, c \in \mathbb{R}^d\}\ \ \Pi_{A\Theta + c} = \Pi_\Theta \circ A^T.$$

This implies that Π_Θ inherits any symmetries (such as invariance under some rotations and/or reflections) that Θ might have. This also means that Π_Θ is independent of the choice of origin. In conjunction with **(5)**, it also implies that Π_Θ is independent of scaling of Θ. Hence, altogether

$$\Pi_{r\Theta + c} = \Pi_\Theta \qquad \forall r \neq 0, c \in \mathbb{R}^d.$$

Finally, each $p \in \Pi_\Theta$ is constant along any lines orthogonal to the affine hull of Θ, i.e.,

$$\Pi_\Theta \subseteq \Pi(\operatorname{affine}(\Theta)),$$

with

$$\operatorname{affine}(\Theta) := \{\sum_{\vartheta \in \Theta} \vartheta\, w(\vartheta) : \sum_{\vartheta} w(\vartheta) = 1\}.$$

(7) minimal degree, i.e., the elements of Π_Θ have as small a degree as is possible. Here is the precise description: For any polynomial space P for which $\langle\Theta, P\rangle$ is correct, and for all j, $\dim P \cap \Pi_j \le \dim \Pi_\Theta \cap \Pi_j$. This implies, e.g., that if $\langle\Theta, \Pi_k\rangle$ is correct, then $\Pi_\Theta = \Pi_k$. In other words, in the most heavily studied case, viz. of Θ for which Π_k is an acceptable choice, our assignment would also be Π_k.

(8) monotonicity, i.e., $\Theta \subset \Theta' \implies \Pi_\Theta \subset \Pi_{\Theta'}$. This makes it possible to develop a Newton form for the interpolant. Also, in conjunction with **(7)** and **(9)**, this ties our scheme closely to standard choices.

(9) Cartesian product $\implies$ tensor product, i.e., $\Pi_{\Theta\times\Theta'} = \Pi_\Theta \otimes \Pi_{\Theta'}$. In this way, our assignment in the case of a rectangular grid coincides with the assignment standard for that case. In fact, in conjunction with **(8)**, we can conclude that we obtain the standard assignment even in the case that Θ is a 'lower' set of a rectangular grid of points (see Section 7).

(10) associated differential operators. This unusual property links polynomials p which vanish on Θ to homogeneous constant coefficient differential operators $q(D)$ which vanish on Π_Θ. The precise statement is that such $q(D)$ vanishes on Π_Θ if and only if the homogeneous polynomial q is the leading term $p_\uparrow$ of some polynomial p which vanishes on Θ. We expect this property to play a major role in formulae for the interpolation error.

(11) constructible, i.e., a basis for Π_Θ can be constructed in finitely many arithmetic steps.

This list provides enough details to make it possible to identify Π_Θ in certain simple situations directly, without the aid of the defining formula (1.2). For example, if $\#\Theta = 1$, then necessarily $\Pi_\Theta = \Pi_0$ (by **(7)**). If $\#\Theta = 2$, then, by **(6)** and **(7)**, necessarily $\Pi_\Theta = \Pi_1(\mathrm{affine}(\Theta))$. 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 have again

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

but only for $k = 1, 3$. When $\mathrm{affine}(\Theta)$ is a plane, we may use **(6)** 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 **(2)** and **(6)** impose further restrictions, it seems possible to construct a suitable map $\theta \mapsto q$ in many ways so that the resulting $\Theta \mapsto \Pi_\Theta$ satisfies all the above conditions, except conditions **(8)** and **(10)** perhaps. (See Section 6 for our choice for $q = q_\theta$.) At present, we do not know whether there is only one map $\Theta \mapsto \Pi_\Theta$ satisfying all conditions **(1)**-**(9)**. But, addition of condition **(10)** uniquely determines the map.

Of course, we didn't make up the above list and then set out to find the map $\Theta \mapsto \Pi_\Theta$. Rather, we came across the fact that the pair $\langle\Theta, (\exp_\Theta)_\downarrow\rangle$ is always correct, and this started us off studying the assignment $\Pi_\Theta := (\exp_\Theta)_\downarrow$.

2. The choice of P provided by elimination

In this section, we provide further insight into our particular assignment $\Pi_\Theta = (\exp_\Theta)_\downarrow$ by comparing it with a more straightforward assignment which is provided by Gauss elimination applied to the Vandermonde matrix for Θ. This also should help in the understanding of the algorithm for the construction of I_Θ described in the next section.

In the absence of bases for the space

$$\Pi := \Pi(\mathbb{R}^d)$$

of all polynomials in d variables more suitable for calculations with multivariable polynomials, we deal here with the **power form**, i.e., we express polynomials as linear combinations of the powers

$$()^\alpha : \mathbb{R}^d \to \mathbb{R} : x \mapsto x^\alpha := x_1^{\alpha(1)} \cdots x_d^{\alpha(d)}.$$

The polynomial $p =: \sum_\alpha ()^\alpha c(\alpha)$ on $\mathbb{R}^d$ matches the function f at the pointset Θ if and only if its coefficient sequence $c := \big(c(\alpha)\big)_{\alpha\in\mathbb{Z}_+^d}$ solves the linear system

$$V? = f_{|\Theta}, \tag{2.1}$$

with

$$V \;:=\; \Big(\vartheta^\alpha\Big)_{\vartheta\in\Theta,\alpha\in\mathbb{Z}_+^d} \tag{2.2}$$

the **Vandermonde matrix for** Θ. Thus a search for polynomial interpolants to f at Θ is a search for solutions $c : \mathbb{Z}_+^d \to \mathbb{R}$ of (2.1) of *finite* support (i.e., with all but finitely many entries equal to zero).

Actual calculations would force us to order the points in Θ and the indices $\alpha \in \mathbb{Z}_+^d$. It is more convenient, though, to let the $\vartheta \in \Theta$ and the $\alpha \in \mathbb{Z}_+^d$ index themselves for the time being. Thus V is a linear map taking functions on $\mathbb{Z}_+^d$ to functions on Θ. Its columns correspond to $\alpha \in \mathbb{Z}_+^d$, its rows to $\vartheta \in \Theta$.

(2.3)Proposition. *The Vandermonde matrix V (see (2.2)) is of full rank.*

Proof: One way to see this is to observe that $(a(\vartheta))_{\vartheta\in\Theta}V = 0$ implies that $\sum_{\vartheta\in\Theta} a(\vartheta)e_\vartheta = 0$ (since $\vartheta^\alpha = (D^\alpha e_\vartheta)(0)$), and thus to rely on the following

(2.4)Fact. *Any collection of exponentials with distinct frequencies is linearly independent.*

Proof: The proof is by induction since the linear independence is obvious when $\#\Theta = 1$. If $\#\Theta > 1$ and $s := \sum_{\vartheta\in\Theta} a(\vartheta)e_\vartheta = 0$ with all $a(\vartheta) \neq 0$, then also $(D_y - c)s = \sum_{\vartheta\in\Theta}((y\cdot\vartheta) - c)a(\vartheta)e_\vartheta = 0$ for any particular y. Since the ϑ are distinct, we can choose y and c so that $y\cdot\theta = c$ for a particular $\theta \in \Theta$ while $y\cdot\vartheta \neq c$ for at least one $\vartheta \in \Theta$. Thus $\sum_{\vartheta\neq\theta}((y\cdot\vartheta) - c)a(\vartheta)e_\vartheta = 0$ is a sum of the same nature but with one fewer summand, hence with all its coefficients zero by induction hypothesis, hence at least one of the $a(\vartheta)$ must be zero, contrary to our assumption. ♠

Elimination is the standard tool provided by Linear Algebra for the determination of the solution set of any linear (algebraic) system. Elimination classifies the unknowns into **bound** and **free**. Assuming the coefficient matrix to be of full rank (which our matrix V is by (2.3)Proposition), this means that each row is designated a pivot row for some unknown, which thereby is "bound", i.e., computable once all "later" unknowns are determined. Any unknown not bound is "free", i.e., freely choosable. Standard elimination proceeds in order, from left to right and from top to bottom, if possible. In Gauss elimination with partial pivoting, one insists on proceeding from left to right, but is willing to rearrange the rows, if necessary. Thus, Gauss elimination with partial pivoting applied to (2.1) (written according to some ordering of the $\vartheta \in \Theta$ and the $\alpha \in \mathbb{Z}_+^d$) produces a factorization

$$LW = V,$$

with L unit lower triangular and W in row echelon form. This means that there is a sequence $\beta_1, \beta_2, \ldots, \beta_n$ which 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.

(2.5)Proposition. *Let $LW = V$ be the factorization of V provided by Gauss elimination with partial pivoting. Specifically, let $\beta_1, \beta_2, \ldots, \beta_n$ be the sequence, strictly increasing in the same total ordering of $\mathbb{Z}_+^d$ that was used to order the columns of V, for which, 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. Then $P := \mathrm{span}((\,)^{\beta_j})_{j=1}^n$ is correct for interpolation at Θ. Moreover, if the columns of V are ordered by degree, then P is a polynomial space of smallest possible degree which is correct for Θ.*

Proof: By assumption, the square matrix

$$U := \Big(W(\vartheta_i, \beta_j)\Big)_{i,j=1}^n$$

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

$$a := (LU)^{-1}\,(f(\vartheta_1), \ldots, f(\vartheta_n)) \tag{2.6}$$

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

Now recall that Gauss elimination determines the next pivot column as the closest possible column to the right of the present pivot column. This means that each β_j is chosen as the smallest possible index greater than β_{j-1}, in whatever order we chose to write down the columns of V. Consequently, the polynomial space

$$P := \mathrm{span}\left((\,)^{\beta_i}\right)_{i=1}^n$$

selected by this process is spanned by monomials of smallest possible exponent (in the ordering of $\mathbb{Z}_+^d$ used). In particular, assume that we ordered the α by **degree**, i.e., so that

$$\alpha < \beta \implies |\alpha| \le |\beta|,$$

with

$$|\alpha| := \alpha(1) + \ldots + \alpha(d)$$

the customary abbreviation for the **length** of the index vector α. Then P is of smallest degree (since elimination applied to a matrix of full rank determines the shortest initial segment of full rank of that matrix). ♠

Now note that the polynomial space P constructed in the proposition may well change drastically in response to a simple change of variables. For example, if $\Theta = \{(0,0),(1,0)\} \subset \mathbb{R}^2$, and we use the standard ordering

$$(0,0),(1,0),(0,1),(2,0),(1,1),(0,2),\ldots$$

for $\mathbb{Z}_+^2$, then, for any rotate $A\Theta$ of Θ, elimination would provide the space $\mathrm{span}\{()^0,()^{1,0}\}$, except for rotation by 90 degrees, in which case $\mathrm{span}\{()^0,()^{0,1}\}$ would be selected.

This simple example also illustrates that the sequence $\beta_1,\beta_2,\ldots,\beta_n$ need not turn out to consist of consecutive terms, even if we ordered the α by $|\alpha|$. (Facts like this have 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 as its goal the 'smallness' of the factors L and U, so the additional freedom of column pivoting allowed here provides further means for keeping the factors L and U 'small'. The smaller these factors, the better is the condition of the corresponding basis $\big(()^{\beta_j}\big)$ for the polynomial space P selected, when considered as a space of functions on Θ.

(2.7)Theorem. *Assume the columns of the Vandermonde matrix $V = \big(\vartheta^\alpha\big)$ ordered by degree and let $LW = V$ be the factorization of V provided by Gauss elimination with partial pivoting. Specifically, let $\beta_1,\beta_2,\ldots,\beta_n$ be the strictly increasing sequence for which, 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. Then $h_i := \sum_{|\alpha|=|\beta_i|} W(\vartheta_i,\alpha)()^\alpha/\alpha!$, $i=1,\ldots,n$ is basis for Π_Θ.*

Proof: Since $V(\vartheta,\alpha) = \vartheta^\alpha = (D^\alpha e_\vartheta)(0)$, and $W = L^{-1}V$, it follows that each $g_i := \sum_\alpha W(\vartheta_i,\alpha)()^\alpha/\alpha!$ is in $\exp_\Theta$, hence $h_i := \sum_{|\alpha|=|\beta_i|} W(\vartheta_i,\alpha)()^\alpha/\alpha! = g_{i\downarrow}$ is in $\Pi_\Theta = (\exp_\Theta)_\downarrow$. Since W is in row-echelon form (specifically, $W(\vartheta_i,\beta_i)$ is the first nonzero entry in the row $W(\vartheta_i,:)$ and the sequence β_i is strictly increasing), we know that $h_1,h_2,\ldots,h_n$ is linearly independent, hence a basis for Π_Θ, since $\dim\Pi_\Theta = n$ by (1.1). ♠

Thus, the assignment $\Theta \mapsto \Pi_\Theta$ proposed in [3] turns out to differ from the naive assignment made in (2.5)Proposition in only one (important) detail: Instead of the space spanned by the particular monomials $()^{\beta_i}$ singled out by elimination, we take the space spanned by the least terms $g_{i\downarrow}$ of the functions $g_i := \sum_\alpha W(\vartheta_i,\alpha)()^\alpha/\alpha!$. But these particular $g_1,g_2,\ldots,g_n$ do not in general satisfy (1.7). To obtain a basis $g_1,g_2,\ldots,g_n$ for $\exp_\Theta$ satisfying (1.7), we carry out elimination, not monomial by monomial, but degree by degree.

3. Elimination by degree

We proposed in [3] a particular algorithm for the construction of the basis $g_1, g_2, \ldots, g_n$ for $\exp_\Theta$ satisfying (1.7) and needed for (1.6). But, with the details of Gauss elimination recalled in the preceding section in mind, it seems more efficient to construct (as already proposed in [2]) the g_i by applying Gauss elimination with partial pivoting to the matrix

$$\mathbf{V} := \left(\vartheta^k\right)$$

obtained from the Vandermonde $V = \left(\vartheta^\alpha\right)$ by treating all entries of a given degree as one entry. We have written $\mathbf{V}$ instead of V to signify this alternate point of view. Thus $\mathbf{V}$ has its rows indexed by $\vartheta \in \Theta$ as before, but its columns are indexed by $k = 0, 1, 2, \ldots$. Correspondingly,

$$\vartheta^k \;=\; \left(\vartheta^\alpha\right)_{|\alpha|=k}.$$

Since the entries of $\mathbf{V}$ are vectors, rather than just numbers, we cannot hope to 'eliminate entries', we can only hope to make all the entries in the pivot column below the pivot row **orthogonal** to the pivot entry. Because of (1.3), the relevant scalar product is

$$\langle a, b\rangle_k := \sum_{|\alpha|=k} a(\alpha)b(\alpha)/\alpha! \tag{3.1}$$

when eliminating in column k of $\mathbf{V}$. In order to keep the notation uncluttered, we will use the abbreviation

$$\langle\!\langle \mathbf{W}(\vartheta, k), \mathbf{W}(\vartheta, k)\rangle\!\rangle := \langle \mathbf{W}(\vartheta, k), \mathbf{W}(\vartheta, k)\rangle_k,$$

with $\mathbf{W}$ any matrix which, like $\mathbf{V}$, has vectors indexed by $\{\alpha \in \mathbb{Z}^d : |\alpha| = k\}$ as the entries in its kth column.

It follows that a given column may be pivot column for several pivot rows. Still, the overall process of Gauss elimination with partial pivoting applied to a matrix like $\mathbf{V}$ is clear: Let $\mathbf{W}$ be the 'working array' which initially equals $\mathbf{V}$. At the jth step, we look for the smallest $k_j \ge k_{j-1}$ for which there is a nontrivial entry of $\mathbf{W}$ in column k_j at or below row j. Then we find a largest such entry (relative to the size of the corresponding entry or row of $\mathbf{V}$) and, if necessary, interchange its row with row j of $\mathbf{W}$ to bring it into the pivot position $\mathbf{W}(\vartheta_j, k_j)$. Then we subtract the appropriate multiple of the pivot row $\mathbf{W}(\vartheta_j, :)$ from all subsequent rows in order to make $\mathbf{W}(\vartheta_i, k_j)$ orthogonal to $\mathbf{W}(\vartheta_j, k_j)$ for all $i > j$.

The result is a factorization

$$L\mathbf{W} = \mathbf{V},$$

with L again unit lower triangular, but $\mathbf{W}$ is in row echelon form in the following sense. There is a nondecreasing sequence $k_1, k_2, \ldots, k_n$ and some ordering $\{\vartheta_1, \vartheta_2, \ldots, \vartheta_n\}$ of Θ so that, for all j, the (vector-)entry $\mathbf{W}(\vartheta_j, k_j)$ is the first nonzero entry in the row $\mathbf{W}(\vartheta_j, :)$ of $\mathbf{W}$. In other words, the matrix

$$\left(\mathbf{W}(\vartheta_i, k_j)\right)_{i,j=1}^{n}$$

is block upper triangular, with nonzero diagonal entries. Note that this matrix need not be upper triangular, since the sequence $k_1, k_2, \ldots, k_n$ need not be strictly increasing. But, there has to be orthogonality of $\mathbf{W}(\vartheta_i, k_j)$ to $\mathbf{W}(\vartheta_j, k_j)$ when $k_i = k_j$ and $i \neq j$. Explicitly, the square matrix

$$U := \Big(\langle\!\langle \mathbf{W}(\vartheta_i, k_j), \mathbf{W}(\vartheta_j, k_j)\rangle\!\rangle\Big)_{i,j=1}^{n} \tag{3.2}$$

is upper triangular and invertible. Consequently, with

$$U\mathbf{G} := \mathbf{W},$$

the matrix

$$\Big(\langle\!\langle \mathbf{G}(\vartheta_i, k_j), \mathbf{G}(\vartheta_j, k_j)\rangle\!\rangle\Big)_{i,j=1}^{n}$$

is diagonal and invertible. For, factoring out the upper triangular matrix U is equivalent to 'backward elimination', i.e., to the calculations

for $j = n, n-1, \ldots, 1$, do:
  $W(\vartheta_j, :) \longleftarrow W(\vartheta_j, :)/U(j,j)$
  for $i = 1, \ldots, j-1$, do:
    $W(\vartheta_i, :) \longleftarrow W(\vartheta_i, :) - U(i,j)W(\vartheta_j, :)$
  end
end

in which the jth step enforces orthogonality of the pivot element in row j to the elements *above* it in the pivot column, without changing the orthogonalities already achieved in subsequent columns, and without changing anything in the preceding columns. Thus, in terms of the weighted scalar product

$$\langle a, b\rangle := \sum_k \langle a, b\rangle_k = \sum_\alpha a(\alpha)b(\alpha)/\alpha! \tag{3.3}$$

for *sequences* $a, b : \mathbb{Z}_+^d \to \mathbb{R}$, we have

$$\langle G(\vartheta_i, :), G_{k_j}(\vartheta_j, :)\rangle = \delta_{i,j}/U(j,j), \quad i, j = 1, \ldots, n, \tag{3.4}$$

with G_k given by

$$G_k(:, \alpha) := \begin{cases} G(:, \alpha), & |\alpha| = k; \\ 0, & \text{otherwise.} \end{cases}$$

With this, let

$$g_i := \sum_\alpha ()^\alpha/\alpha! \;\; G(\vartheta_i, \alpha). \tag{3.5}$$

Then

$$\sum_j (LU)(i,j) g_j = e_{\vartheta_j}, \quad \text{all } j, \tag{3.6}$$

since $LUG = V = \big(D^\alpha e_{\vartheta_i}(0)\big)_{i,\alpha}$. Further,

$$g_{i\downarrow} = \sum_{|\alpha|=k_i} ()^\alpha/\alpha!\ G(\vartheta_i,\alpha) \quad = \quad \sum_\alpha ()^\alpha/\alpha!\ G_{k_i}(\vartheta_i,\alpha), \tag{3.7}$$

and we conclude from (3.4) that

$$\langle g_i, g_{j\downarrow}\rangle = \delta_{i,j}/U(j,j), \quad i,j=1,\ldots,n. \tag{3.8}$$

Since $g_1, g_2, \ldots, g_n$ is linearly independent by (3.8), (3.6) implies that $g_1, g_2, \ldots, g_n$ is a basis for $\exp_\Theta$. But (3.8) also implies that $g_{1\downarrow}, \ldots, g_{n\downarrow}$ so constructed is linearly independent, hence a basis for Π_Θ by (1.1).

This proves

(3.9)Theorem. *The functions g_i defined by (3.5) provide a basis for $\exp_\Theta$ which satisfies (1.7), and the corresponding $g_{i\downarrow}$ form a basis for Π_Θ.*

(3.10)Corollary. *Let*

$$a := \operatorname{diag}(U)(LU)^{-1}\ (f(\vartheta_1), \ldots, f(\vartheta_n)), \tag{3.11}$$

with L, U, and $\vartheta_1, \vartheta_2, \ldots, \vartheta_n$ determined during Gauss elimination with partial pivoting applied to $\mathbf{V}$ as described above. Then, with $g_{i\downarrow}$ as given by (3.7),

$$I_\Theta f = \sum_j g_{j\downarrow} a(j)$$

is the unique interpolant from Π_Θ to f on Θ.

Proof: The function $q := \sum_j g_{j\downarrow} a(j)$ is in Π_Θ by (3.9)Theorem. Further, from (3.8), $\langle g_j, q\rangle = a(j)/U(j,j)$. Therefore, from (3.6) and (3.11),

$$q(\vartheta_i) = \langle e_{\vartheta_i}, q\rangle = \sum_j LU(i,j)\langle g_j, q\rangle = \sum_j LU(i,j) \sum_r (LU)^{-1}(j,r) f(\vartheta_r) = f(\vartheta_i).$$

In effect, the multiplication in (3.11) by the diagonal matrix $\operatorname{diag}(U)$ accounts for the division by $\langle g_j, g_{j\downarrow}\rangle$ in (1.6), as the latter number is $1/U(j,j)$, by (3.8). ♠

It is worth noting that the factoring out of U from $\mathbf{W}$ will not change the pivot entries $\mathbf{W}(\vartheta_j, k_j)$ since $U(i,j) = 0$ if $k_i = k_j$ and $i \neq j$, except for the normalizing division. In other words,

$$G_{k_i} = W_{k_i}/U(i,i),$$

(with W_k defined entirely analogously to G_k), showing that the factoring out of U from W need not be carried out, unless one is interested in the g_i rather than the $g_{i\downarrow}$. On the other hand, formation of U is essential for the calculation of the coefficients of the interpolating polynomial.

In the language introduced in this section, the algorithm for the calculation of suitable $g_1, g_2, \ldots, g_n$ from $f_j := e_{\vartheta_j}$, $j = 1, \ldots, n$ proposed in [3] amounts to Gauss elimination with *column* pivoting applied to $\mathbf{V}$, except that no columns are actually interchanged. Rather, at the jth step, one looks for the left-most nonzero entry in the jth row of the working array $\mathbf{W}$, say the entry $\mathbf{W}(\vartheta_j, k_j)$, then uses the jth row to make all entries $\mathbf{W}(\vartheta_i, k_j)$ for $i \neq j$ orthogonal to $\mathbf{W}(\vartheta_j, k_j)$. This will not spoil orthogonality of $\mathbf{W}(\vartheta_i, k_i)$ to $\mathbf{W}(\vartheta_r, k_i)$ for $r \neq i$ and $i < j$ achieved earlier, since either $k_i < k_j$, hence $\mathbf{W}(\vartheta_j, k_i) = 0$, or $k_i > k_j$, hence $\mathbf{W}(\vartheta_j, k_j)$ is trivially orthogonal to $\mathbf{W}(\vartheta_i, k_j) = 0$, or $k_i = k_j$, hence $\mathbf{W}(\vartheta_j, k_j)$ is already orthogonal to $\mathbf{W}(\vartheta_i, k_j)$. Thus one obtains a factorization

$$A\mathbf{W} = \mathbf{V},$$

with A invertible, and $\mathbf{W}$ in reduced row echelon form in the sense that, for some sequence $k_1, k_2, \ldots, k_n$,

$$\langle \mathbf{W}(\vartheta_i, k_j), \mathbf{W}(\vartheta_j, k_j) \rangle_{k_j} = 0 \quad \Longleftrightarrow \quad i \neq j.$$

This implies that the functions $g_1, g_2, \ldots, g_n$ defined by

$$g_j := \sum_\alpha ()^\alpha / \alpha! \;\; W(\vartheta_j, \alpha)$$

satisfy (1.7), hence the corresponding $g_{j\downarrow}$ must be a basis for Π_Θ (by the reasoning used earlier). It is not obvious without recourse to the results from [3] that the two sequences $g_{1\downarrow}, \ldots, g_{n\downarrow}$ produced by the two algorithms span the same space.

The algorithm outlined in this section seems preferable to the one from [3] not only because it is closer to a standard algorithm but also because it provides a ready means for ordering the points of Θ for greater stability of the calculations.

Some of the finer computational details are taken up below, after a short section on a particularly suitable polynomial form.

4. Nested multiplication for the modified power form

We know only two polynomial forms readily available for the presentation of polynomials in several variables, the *power form* and the *Bernstein-Bézier form*. The calculations above are in terms of the **power form**

$$p = \sum_\alpha ()^\alpha \; D^\alpha p(0)/\alpha!,$$

hence we stick with that form here, particularly since we are not concerned here with the Bernstein-Bézier form's major strength, the smooth patching of polynomial pieces (see, e.g., [1]).

It is only prudent to use the **shifted** power form, i.e., to write

$$p = \sum_\alpha (\cdot - c)^\alpha \; D^\alpha p(c)/\alpha!,$$

for some appropriate **center** c, e.g.,

$$c = c_\Theta := \sum_{\vartheta\in\Theta} \vartheta/\#\Theta.$$

Equivalently, we assume that Θ has been shifted at the outset by its center c_Θ.

It turns out to be simpler to use the following **modified power form**

$$p = \sum_\alpha ()^\alpha \binom{|\alpha|}{\alpha}\ D^\alpha p(0)/|\alpha|!\ , \tag{4.1}$$

with $\binom{|\alpha|}{\alpha} := |\alpha|!/\alpha!$ the multinomial coefficients. There are two reasons.

(i) It is easy to program and use the following multivariable version of **nested multiplication** (or **Horner's scheme**):

(4.2)Proposition. *If*

$$c(\alpha) := \begin{cases} D^\alpha p(0)/|\alpha|!, & |\alpha| = \deg p; \\ D^\alpha p(0)/|\alpha|! + \sum_{i=1}^d x_i c(\alpha + \mathbf{i}_i), & |\alpha| = \deg p - 1, \deg p - 2, \ldots, 0, \end{cases} \tag{4.3}$$

with $\mathbf{i}_i$ the ith unit vector, $p \in \Pi(\mathbb{R}^d)$, and $x \in \mathbb{R}^d$, then $c(0) = p(x)$.

Proof: Indeed, it follows that

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

with n_α the number of different increasing paths to α from the origin through points of $\mathbb{Z}_+^d$. This number is $n_\alpha = \binom{|\alpha|}{\alpha}$, hence $c(0) = p(x)$, by (4.1). ♠

In effect, it is possible to evaluate a multivariable polynomial p from its normalized Taylor coefficients $(D^\alpha p)(0)/|\alpha|!$ without the (explicit) computation of multinomial coefficients.

(ii) The information about g_j computed by the algorithm outlined in the preceding section readily provides the numbers $D^\alpha g_j(0)$ (see (3.5)), hence the calculation of the modified power form for $I_\Theta f$, i.e., of the normalized Taylor coefficients $(D^\alpha I_\Theta f)(0)$, from the matrix G and the vector $(LU)^{-1} f_{|\Theta}$ can be accomplished without generation and use of the multinomial coefficients.

The close similarity to de Casteljau's algorithm (see, e.g., [1]) for the evaluation of the Bernstein-Bézier form is actually not surprising, for the following reason. The **Bernstein-Bézier form**

$$x \mapsto \sum_{|\beta|=k} (\xi(x))^\beta \binom{|\beta|}{\beta}\ c(\beta)$$

describes a polynomial of degree $\le k$ in terms of the $d+1$ linear polynomials ξ_t defined by the identity

$$\sum_{t\in T} \xi_t p(t) = p \quad \text{ for all } p \in \Pi_1,$$

with $T \subset \mathbb{R}^d$ in general position. The de Casteljau algorithm for its evaluation at some x consists of the calculations

$$c(\beta) := \sum_{t \in T} \xi_t(x) c(\beta + \mathbf{i}_t), \qquad |\beta| = j,$$

for $j = k-1, k-2, \ldots, 0$, with the resulting $c(0)$ the desired value at x. While the vector $\xi(x)$ provides the barycentric coordinates of x with respect to the pointset T, no use is made in the de Casteljau algorithm of the fact that $\sum_{t \in T} \xi_t(x) = 1$. Thus the calculations

$$c(\alpha) := \sum_{i=1}^{d} x_i c(\alpha + \mathbf{i}_i), \qquad |\alpha| = j,$$

for $j = k-1, k-2, \ldots, 0$ and started from given $c(\alpha)$ with $|\alpha| = k$ will provide the number

$$\sum_{|\alpha|=k} x^\alpha \binom{|\alpha|}{\alpha} c(\alpha).$$

The full algorithm above merely combines appropriately the steps common to de Casteljau applied to the terms in (4.1) of different degrees.

5. Algorithmic details

We give here a (somewhat informal) MATLAB-like program (see, e.g., [6] for language details) for the construction of our interpolant in order to document the simplicity of the actual calculations needed. In this 'program', we use the following conventions:

`V` and `W` denote the matrices $\mathbf{V}$ and $\mathbf{W}$, respectively. In particular, `W(i,k)` is a vector with $\binom{\mathtt{k}+d-1}{d-1}$ entries, indexed by $\{\alpha \in \mathbb{Z}^d : |\alpha| = \mathtt{k}\}$. This is decidedly not allowable in present-day MATLAB, but convenient here, as it avoids discussion of the (important technical) question of the best way to order the index set $\{\alpha \in \mathbb{Z}^d : |\alpha| = \mathtt{k}\}$.

Correspondingly, for two vectors `a` and `b` (such as `W(i,k)`, `W(j,k)`) indexed by $\{\alpha \in \mathbb{Z}^d : |\alpha| = \mathtt{k}\}$, `<a,b>` denotes the (scaled) scalar product

$$\mathtt{<a,b>} := \sum_{|\alpha|=\mathtt{k}} \mathtt{a}(\alpha)\mathtt{b}(\alpha)\binom{|\alpha|}{\alpha} \tag{5.1}$$

related to (3.1) (with $k = \mathtt{k}$). All matrices mentioned in the 'program' other than `V` and `W` are proper MATLAB matrices, i.e., have scalar entries.

Further, we use `a <-- b` to indicate that `a` is to be overwritten with the contents of `b`, and use an occasional English word or two to describe an action whose details seem clear.

We borrow from MATLAB the notations: (i) `eye(n,n)` for the identity matrix of order `n`; (ii) `ones(m,n)` for the matrix of size $\mathtt{m} \times \mathtt{n}$ with all entries equal to 1; (iii) `zeros(m,n)` for the matrix of size $\mathtt{m} \times \mathtt{n}$ with all entries equal to 0; (iv) `a:b` for the vector with entries

$\mathtt{a}, \mathtt{a}+1, \ldots, \mathtt{a}+m$, with m the natural number for which $\mathtt{a}+m \le \mathtt{b} < \mathtt{a}+m+1$; (v) `A*B` for the matrix product of the matrices `A` and `B`; (vi) standard logical constructs like (`for j=1:n,...,end`), and (`if ..., ..., end`); (vii) the construct (`while 1, ..., if ..., break, end, ..., end`), which is a loop exited only through the `break`; (viii) the construct `[m,i] <-- max(a)` to provide $\mathtt{m} := \mathtt{a}(\mathtt{i}) := \max_j \mathtt{a}(j)$.

```
% INPUT: Θ = {ϑ_1,...,ϑ_n}, f_|Θ, tol
k <-- 0
V(:,k) <-- ones(n,1);  W(:,k) <-- V(:,k)
L <-- eye(n,n); U <-- zeros(n,n); K <-- zeros(1,n)
for j=1:n
   while 1
      [m,i] <-- max{<W(i,k),W(i,k)>/<V(i,k),V(i,k)>:  i>j-1}
      if m>tol, break, end
      k <-- k+1
      V(:,k) <--  from V(:,k-1) and Θ
      W(:,k) <-- L^(-1)*V(:,k)
   end
   if i>j, interchange rows i and j, end
   K(j) <-- k
   for i=1:j,
      U(i,j) <-- <W(i,k),W(j,k)>
   end
   for i=j+1:n
      L(i,j) <-- <W(i,k),W(j,k)>/U(j,j)
      W(i,k) <-- W(i,k) - L(i,j)*W(j,k)
   end
end
W <-- U^(-1)*W
%
f <-- properly permuted f_|Θ
a <-- diag(U)*U^(-1)*L^(-1)*f
kmax <-- max(K); dk <-- ( [0:kmax]+d-1 choose d-1 )
coefs <-- zeros(kmax+1,dk(kmax))
for j=1:n
  range=1:dk(K(j))
  coefs(K(j),range) <-- coefs(K(j),range) + a(j)*W(j,K(j))
end
% OUTPUT: coefs
```

The output provides the coefficients $c(\alpha) = \mathtt{coefs}(|\alpha|, \alpha)$ for the modified power form of the interpolant. The needed weights $w(\alpha) := \binom{|\alpha|}{\alpha}$ for the (scaled) scalar product (5.1) are integers and are conveniently generated from their recurrence relation

$$w(\alpha) = \sum_{i=1}^{d} w(\alpha - \mathbf{i}_i) \tag{5.2}$$

each time k is increased by 1, using the initial value

$$w(0) = 1$$

and the side conditions

$$w(\alpha) = 0 \qquad \alpha \notin \mathbb{Z}_+^d.$$

The algorithm does require a sensible ordering of the index set $\{\alpha : |\alpha| = k\}$ in order to facilitate (i) use of (5.2); (ii) the efficient calculation of the entries of $\mathbf{V}(\vartheta, k)$ from those of $\mathbf{V}(\vartheta, k-1)$, e.g., via

(5.3) $\quad V(\vartheta, \alpha) = \vartheta_i V(\vartheta, \alpha - \mathbf{i}_i), \qquad$ with α s.t. $\alpha(j) = 0$ for $j < i$; and for $i = 1, \ldots, d$;

and (iii) the use of (4.3). We have used the inverse lexicographic ordering of the α.

The output does depend on the choice of the tolerance `tol`. In exact arithmetic, we could choose `tol` $= 0$.

(5.4)Proposition. *If the above algorithm is run in exact arithmetic with* `tol` $= 0$, *then the* `while`*–loop is never repeated.*

Proof: In exact arithmetic and with `tol` $= 0$, the algorithm provides a homogeneous basis for Π_Θ, by (3.9)Theorem, and the polynomial degrees of these basis elements are the numbers `k` appearing in the 'program'. Under certain conditions, this number is increased by 1 in the `while`–loop. If a second increase were necessary for the current `j`, then it would follow that there is a gap in the degrees of some homogeneous basis for Π_Θ, and this would contradict the D-invariance of Π_Θ (see **(4)**). ♠

In finite-precision arithmetic, the choice of `tol` is more delicate. It should reflect the number of digits carried during the calculations. As `tol` is increased, we can expect L and U to be of smaller size, but may eventually not obtain a polynomial space close to Π_Θ. This can be of considerable numerical advantage in the case that a zero tolerance would lead to an unacceptably large U. This is analogous to the common practice of treating a cluster of (simple) zeros of a function numerically as one zero of appropriate multiplicity.

6. Examples

(6.1)Four points in the plane. We start with the simplest nontrivial Θ, viz. a Θ made up of four points whose affine hull is a plane. As already pointed out in Section 1, we may assume without loss that $\Theta = \{0, \mathbf{i}_1, \mathbf{i}_2, (u, v)\}$, with $\mathbf{i}_j$ the jth unit vector in $\mathbb{R}^2$. With this ordering of the points, and the lexicographic ordering for the α, the Vandermonde matrix becomes

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \ldots \\ 1 & 1 & 0 & 1 & 0 & 0 & \ldots \\ 1 & 0 & 1 & 0 & 0 & 1 & \ldots \\ 1 & u & v & u^2 & uv & v^2 & \ldots \end{pmatrix}.$$

Elimination (without pivoting) with the *scaled* scalar product (5.1) generates the matrices

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & u & v & 1 \end{pmatrix}, \qquad W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & 1 & \dots \\ 0 & 0 & 0 & u^2-u & uv & v^2-v & \dots \end{pmatrix}.$$

Hence

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & u(u-1) \\ 0 & 0 & 1 & v(v-1) \\ 0 & 0 & 0 & w \end{pmatrix}, \qquad w := (u^2+v^2)^2 - 2(u^3+v^3) + (u^2+v^2).$$

This gives

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 1-u^2(u-1)^2/w & -u^2(u-1)v/w & -u(u-1)v(v-1)/w & \dots \\ 0 & 0 & 1 & -u(u-1)v(v-1)/w & -uv^2(v-1)/w & 1-v^2(v-1)^2/w & \dots \\ 0 & 0 & 0 & u(u-1)/w & uv/w & v(v-1)/w & \dots \end{pmatrix}.$$

The resulting basis for Π_Θ consists of

$$g_{1\downarrow} = ()^0, \quad g_{2\downarrow} = ()^{1,0}, \quad g_{3\downarrow} = ()^{0,1},$$

$$g_{4\downarrow} = ()^{2,0}u(u-1)/(2w) + ()^{1,1}uv/w + ()^{0,2}v(v-1)/(2w),$$

with $\langle g_i, g_{4\downarrow}\rangle = (1/w)\delta_{i4}$. This illustrates the fact that, for the purpose of constructing the interpolant, there is no need to construct the matrix G.

Even for this simple example, the resulting formulae are not particularly simple or pretty. On the other hand, there is no suggestion here to carry out such calculations by hand. On the third hand, it is easy to see in this simple example what happens as points become collinear. E.g., if $v \to 0$, $g_{4\downarrow}$ simplifies to

$$g_{4\downarrow} = ()^{2,0}/2(u(u-1)),$$

i.e., Π_Θ now contains $\Pi_2(\mathbb{R} \times \{0\})$, as it should. But this works out only if $u \notin \{0,1\}$. If $(u,v) \to 0$ or $(u,v) \to (1,0)$, then the fourth point (u,v) would be approaching the first or second point in Θ and the limiting Π_Θ now will depend on just how this approach is made.

(6.2)Hexagon points. Because of the inherent symmetries, the formula for interpolation at the vertices of a regular hexagon is very pretty indeed. Assume without loss that

$$\Theta = \{\vartheta_j := (\cos(t_j), \sin(t_j)) : j = 1, \dots, 6\},$$

with $t_j := 2\pi j/6$, all j.

Since $\dim \Pi_2(\mathbb{R}^2) = 6$, we expect $\Pi_\Theta = \Pi_2(\mathbb{R}^2)$ for the generic 6-point set Θ in the plane. But the hexagon points lie on the unit circle, i.e., the polynomial

$$p := 1 - ()^{2,0} - ()^{0,2} \in \Pi_2(\mathbb{R}^2)$$

vanishes on Θ, hence $\langle \Theta, \Pi_2(\mathbb{R}^2) \rangle$ cannot be correct. Further, by **(10)**, $p_\uparrow$ must be orthogonal (in the sense of the pairing (1.3)) to Π_Θ. On the other hand, any five of these six points are generic, i.e., they are not collinear. Hence

$$\Pi_\Theta = (\Pi_2 \ominus \operatorname{span}(p_\uparrow)) + \operatorname{span}(q),$$

with the orthogonal complement $(\Pi_2 \ominus \operatorname{span}(p_\uparrow))$ of $\operatorname{span}(p_\uparrow)$ in Π_2 taken in the sense of the pairing (1.3), and with q a certain homogeneous third-degree polynomial.

Here is one way to determine this q: Consider the interpolant $I_\Theta f$ to data $f(\vartheta_j) = (-)^j$, all j. There are three lines through the origin not containing any of the interpolation points but such that reflection across that line leaves Θ unchanged. By **(6)**, such reflection must also leave Π_Θ unchanged. On the other hand, it will map the data to their negative. This implies that such reflection must map $I_\Theta f$ to its negative, consequently $I_\Theta f$ must vanish along each of these three lines. Therefore, $I_\Theta f$ vanishes to second degree at the origin, hence is a homogeneous cubic, hence $q = I_\Theta f$ (since q is determined only up to scalar multiples, anyway). In particular, rotation by $\pi/3$ maps q to its negative.

Another way is to note that Θ is unchanged under rotation by $\pi/3$, hence such rotation must map q to rq for some real r for which $r^6 = 1$. The choice $r = 1$ would lead to the conclusion that q is constant on Θ, therefore constant throughout, by uniqueness of the interpolant, and this would contradict the fact that q is a third-degree polynomial. Thus, necessarily, $r = -1$, i.e., rotation by $\pi/3$ maps q to its negative. This is the same conclusion reached in the preceding paragraph.

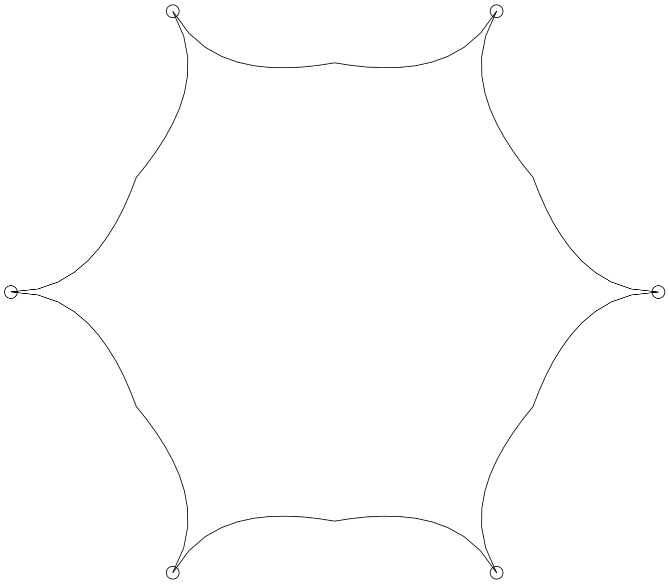
It follows that

$$q = ()^{3,0} - 3()^{1,2} = \operatorname{Re} z^3,$$

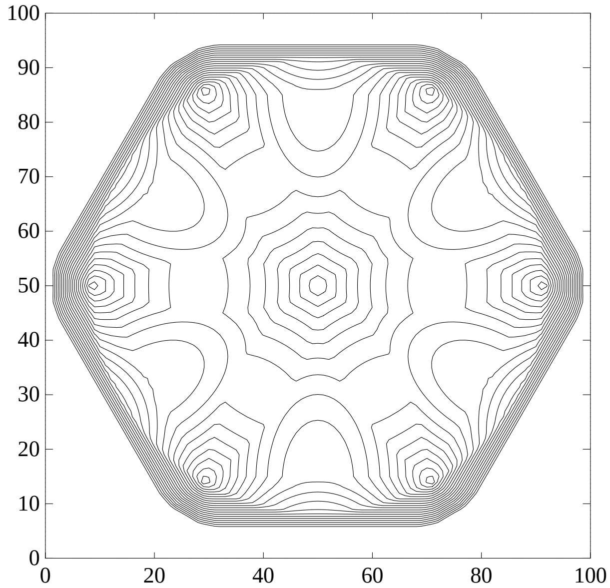
with $z := ()^{1,0} + i()^{0,1}$ the complex independent variable. This is related to the fact that $I_\Theta f$ is the real part of the (complex) Lagrange interpolant to the data $(f(\vartheta_j))_j$ at the six roots of unity in the complex plane. (Since z^3 takes the value $(-)^j$ at the point $\vartheta_j(1) + i\vartheta_j(2)$, $z \mapsto z^3$ is an interpolant from $\Pi_5(\mathbb{C})$, therefore the interpolant).

As a matter of fact, Property **(10)** implies that, for any Θ on any particular circle, Π_Θ must consist of harmonic polynomials since then Θ is mapped to zero by some polynomial whose leading part is $()^{2,0} + ()^{0,2}$. In particular, for *any* six-point set Θ on a circle, the homogeneous cubic polynomial in Π_Θ is a linear combination of $\operatorname{Re} z^3$ and $\operatorname{Im} z^3$.

The three heavy lines in (6.3)Figure show the zeros of the (real) Lagrange polynomial ℓ_6 for our interpolation at the hexagon points. This means that ℓ_6 is the unique polynomial in Π_Θ which is 1 at $(1,0)$ and 0 at the other five points. From the figure (and the fact that $\ell_6(1,0) = 1$), we conclude that ℓ_6 is positive near 0. Since Π_Θ is invariant under rotation by $\pi/3$, the remaining five Lagrange polynomials ℓ_j, $j = 1, \ldots, 5$, are obtained from ℓ_6 by rotation by the appropriate multiple of $\pi/3$, hence are also positive near the origin. Since the Lagrange polynomials sum to the constant function 1, this implies the following surprising fact.



(6.3)Figure. The Lebesgue function for interpolation at the hexagon points is one on the entire central portion.



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

(6.4)Proposition. *In the hexagonal domain outlined by the zerosets of the ℓ_j, the value of the interpolant is a convex combination of the given function values.*

On the other hand, as the radius of the circle shrinks to zero, the interpolant $I_\Theta f$ approximates f near 0 only to first order since the process fails to reproduce all of Π_2. In order to remedy this, we enlarge Θ by adding the origin to it. This is bound to destroy the positivity near the origin of the Lagrange polynomials for the other points (and makes $p = 1 - ()^{2,0} - ()^{0,2}$ the Lagrange polynomial for the seventh point, with the remaining Lagrange polynomials obtained from those for the hexagon points by subtraction of $p/6$). But the Lebesgue function for the resulting interpolation process, i.e., the sum of the absolute values of all the Lagrange polynomials, stays remarkably close to 1 near the origin. (6.5)Figure shows the level lines of the Lebesgue function, for the values 1:.05:2. This shows that, on the unit circle, the Lebesgue function does not exceed 1.5.

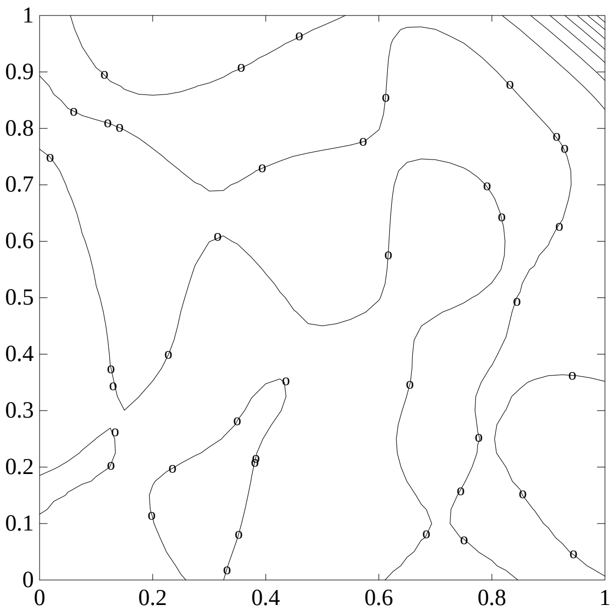


Figure 4. Contour lines for error in interpolation at 40 random points.

(6.6)Random points In this example, we chose 40 points in the square $[0..1]^2$ at random as interpolation points, and constructed an interpolant at these points to the function $x \mapsto \exp(-x_1^2 - x_2^2)$. (6.5)Figure shows ten level lines (corresponding to ten equal values between the minimum and the maximum) of the error in the resulting interpolant. In particular, (6.5)Figure makes clear that the error is close to zero in an area around the interpolation points. The absolutely largest error turned out to be $3e{-}4$. We found that this example was not at all isolated.

7. Verification of list of properties

In this section, we verify the various properties asserted for our interpolation scheme in Section 1.

The first eight properties are either evident from our definition of Π_Θ or are dealt with in detail in [3]. But some of the consequences mentioned still require proof.

We begin with a derivation of the **affine hull property**

$$\Pi_\Theta \subseteq \Pi(\text{affine}(\Theta)).$$

First we give a proof based on **(6)**, and then follow this with shorter proofs based on the stronger properties **(9)** and **(10)**.

After a rigid motion, we may assume that $\text{affine}(\Theta) = \mathbb{R}^s \times \{0\}$. Since Π_Θ is invariant under any linear change of variables which leaves Θ unchanged, it is in particular invariant under any scaling of the arguments $s+1, \ldots, d$. This implies that Π_Θ must contain any polynomial p which does not depend on its arguments $s+1, \ldots, d$ and agrees on $\mathbb{R}^s \times \{0\}$ with some polynomial from Π_Θ. On the other hand, the restriction map $p \mapsto p_{|\mathbb{R}^s \times \{0\}}$ must be 1-1 on Π_Θ, since $\Theta \subset \mathbb{R}^s \times \{0\}$ and the restriction map $p \mapsto p_{|_\Theta}$ is 1-1.

The affine hull property can also be derived from the property **(9)** concerning tensor products, since, after that linear change of variables which makes $\text{affine}(\Theta) = \mathbb{R}^s \times \{0\}$, we may think of Θ as $\Theta = (\Theta \subset \mathbb{R}^s \times \{0\}) \times (\{0\} \subset \mathbb{R}^{d-s})$.

Finally, the affine hull property can also be written

$$D_y(\Pi_\Theta) = 0 \quad \text{for all } y \perp \text{affine}(\Theta),$$

i.e., for all y for which the linear polynomial $x \mapsto y \cdot x$ is constant on Θ. Thus this property is a very special case of **(10)**.

Next, we note that the minimal degree property **(7)** is equivalent to the property

(7') degree-reducing, i.e., for every polynomial p, $\deg I_\Theta p \le \deg p$.

(7.1)Proposition. *Let $\langle \Theta, P \rangle$ be correct, with P a polynomial space, and let I be the corresponding interpolation map. Then P is of minimal degree if and only if* $\deg Ip \le \deg p$ *for all* $p \in \Pi$.

Proof: Assume that $\langle \Theta, P \rangle$ is correct and that $p \in \Pi$. Extend Ip to a **graded** basis B for P, i.e., a B for which $B \cap \Pi_j$ is a basis for $P \cap \Pi_j$ for all j. Since $\langle \Theta, P \rangle$ is correct, it follows that B must be linearly independent over Θ. Since $Ip = p$ on Θ, it follows that $(B \backslash Ip) \cup p$ is also linearly independent over Θ, hence its span, Q say, is also correct for interpolation on Θ. If now $\deg Ip > \deg p =: j$, then it follows that $\dim Q \cap \Pi_j > \dim P \cap \Pi_j$, hence P is not of minimal degree.

Conversely, if $\deg p \ge \deg Ip$ for all polynomials p, then $I(\Pi_k) \subseteq \Pi_k \cap P$, while, for any correct $\langle \Theta, Q \rangle$, I is 1-1 on Q, hence linear independence preserving, and one therefore has $\dim(\Pi_k \cap Q) = \dim I(\Pi_k \cap Q)$, while also $I(\Pi_k \cap Q) \subset \Pi_k \cap P$. Therefore, P has the minimal degree property. ♠

Property **(9)** asserts that $\Pi_{\Theta \times \Theta'} = \Pi_\Theta \otimes \Pi_{\Theta'}$. For its proof, observe that

$$\Pi_{\Theta \times \Theta'} = (\exp_{\Theta \times \Theta'})_\downarrow = (\exp_\Theta \otimes \exp_{\Theta'})_\downarrow,$$

that

$$\Pi_\Theta \otimes \Pi_{\Theta'} = (\exp_\Theta)_\downarrow \otimes (\exp_{\Theta'})_\downarrow,$$

and that

$$(f \otimes g)_\downarrow = (f_\downarrow) \otimes (g_\downarrow),$$

hence that, for any homogeneous basis B for Π_Θ and any homogeneous basis B' for $\Pi_{\Theta'}$, the collection $B \otimes B' := \{b \otimes b' : (b, b') \in B \times B'\}$ is a basis for $\Pi_\Theta \otimes \Pi_{\Theta'}$ and in $\Pi_{\Theta \times \Theta'}$, hence a basis for it since it consists of $\#(B \times B') = \#(\Theta \times \Theta') = \dim \Pi_{\Theta \times \Theta'}$ elements.

For $i = 1, \ldots, d$, let $\Theta_i = \{x_i(0), \ldots, x_i(\gamma(i))\}$ be a collection of $\gamma(i)$ points in $\mathbb{R}$. Inductive application of **(9)** shows that

$$\Pi_\Theta = \bigotimes_{i=1}^{d} \Pi_{\gamma(i)}(\mathbb{R}) \qquad \text{for } \Theta = \times_{i=1}^{d} \Theta_i. \tag{7.2}$$

In conjunction with the monotonicity property **(8)**, this implies that our Π_Θ coincides with the standard assignment even in the case when Θ is an 'order-closed' subset of a rectangular grid

$$\times_i \Theta_i =: \times_{i=1}^{d} \{x_i(0), \ldots, x_i(\gamma(i))\}.$$

Here, we call Θ **order-closed** (or, a 'lower set') if it is of the form $\Theta = \{\theta_\alpha : \alpha \in \Gamma\}$ for $\theta_\alpha := (x_1(\alpha(1)), \ldots, x_d(\alpha(d)))$, with Γ an order-closed subset of C_γ, i.e., $\alpha \in \Gamma$ and $\beta \le \alpha$ implies $\beta \in \Gamma$, and

$$C_\gamma := \{0, \ldots, \gamma(1)\} \times \cdots \times \{0, \ldots, \gamma(d)\}.$$

Thus,

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

We show now that the standard assignment for this Θ, i.e., the space $\operatorname{span}\{()^\beta : \beta \in \Gamma\}$, does indeed coincide with Π_Θ.

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 $\Pi_\alpha := \operatorname{span}\{()^\beta : \beta \le \alpha\}$, by (7.2). By **(8)**, each such Π_α must be contained in Π_Θ, hence $\operatorname{span}\{()^\beta : \beta \in \Gamma\} \subset \Pi_\Theta$, and, since $\dim \Pi_\Theta = \#\Theta = \#\Gamma$, Π_Θ must coincide with that span.

Note that the particular ordering of the points $x_i(j)$ is arbitrary. For example, for each of the following three datapoint distributions Θ

$$\begin{array}{cccccc} x & x & & & & \\ x & x & & & & \\ x & x & & & & \\ x & x & & & & \\ x & x & x & x & x & x \\ x & x & x & x & x & x \end{array} \qquad \begin{array}{cccccc} x & x & x & x & x & x \\ x & & & & & x \\ x & & & & & x \\ x & & & & & x \\ x & & & & & x \\ x & x & x & x & x & x \end{array} \qquad \begin{array}{cccccc} x & x & x & x & x & x \\ x & x & x & x & x & x \\ & & x & x & & \\ & & x & x & & \\ & & x & x & & \\ & & x & x & & \end{array}$$

in the plane, the discussion just given proves that

$$\Pi_\Theta = \Pi_1(\mathbb{R}) \otimes \Pi_5(\mathbb{R}) + \Pi_5(\mathbb{R}) \otimes \Pi_1(\mathbb{R}).$$

For, each set is obtained from a rectangular set of 6×6 points by retaining two rows and two columns. It is not all that hard to see, in each of these figures, the two rectangles, one 2×6 and the other 6×2, which give rise to the corresponding sum of polynomial spaces.

Finally, **(10)** is proved in [5].

8. Generalizations

With minor changes, the entire discussion can be extended to the situation when we consider an arbitrary linearly independent sequence $\lambda_1, \lambda_2, \ldots, \lambda_n$ of linear functionals instead of the particular linear functionals

$$f \mapsto f(\vartheta)$$

with ϑ in some n-set Θ.

Assuming the linear functionals λ_i to be regular enough to be representable as

$$\lambda_i f = \langle f_i, f\rangle \qquad \text{for } f \in \Pi,$$

with f_i functions analytic at the origin, the appropriate Vandermonde-like matrix now has in its ith row the derivatives $D^\alpha f_i(0)$. For small enough x,

$$f_i(x) = \lambda_i(\exp^{\cdot x}).$$

The computations are otherwise unchanged. In particular, the homogeneous polynomials $g_{1\downarrow}, \ldots, g_{n\downarrow}$ constructed span a scale-invariant polynomial space of smallest degree on which the sequence $\lambda_1, \lambda_2, \ldots, \lambda_n$ is maximally linearly independent. Further,

$$\sum_j g_{j\downarrow} \frac{\langle g_j, f\rangle}{\langle g_j, g_{j\downarrow}\rangle} \tag{8.1}$$

is the unique element in that space which agrees with f at the λ_i.

This general setting is discussed in [5] in detail, and also for various specific choices of the 'interpolation conditions' $\lambda_1, \lambda_2, \ldots, \lambda_n$. The discussion there covers even linear functionals which are not 'regular' in the above sense and also the situation when we have infinitely many of them.

Finally, we note that, starting with an arbitrary basis $f_1, f_2, \ldots, f_n$ for some linear subspace H of functions analytic at the origin, the homogeneous polynomials $g_{1\downarrow}, \ldots, g_{n\downarrow}$ provided by the algorithm form a basis for $H_\downarrow = \mathrm{span}\{f_\downarrow : f \in H\}$. In particular, this leads to a construction of a basis for the space of polynomials in the span of the translates of a box spline, as is detailed in [4].

References

[1] C. de Boor, B-form basics, in *Geometric Modeling*, G. Farin ed., SIAM (1987), 131-148.

[2] C. de Boor, Polynomial interpolation in several variables, Proceedings of the *Conference honoring Samuel D. Conte*, R. DeMillo and J.R. Rice eds., Plenum Press, 199x, xxx–xxx.

[3] C. de Boor and A. Ron, On multivariate polynomial interpolation, *Constructive Approximation* **6** (1990), 287–302.

[4] C. de Boor and A. Ron, On ideals of finite codimension with applications to box spline theory, *J.Math.Anal.Appl.* **xx** (199x), xxx–xxx.

[5] C. de Boor and A. Ron, The least solution for the polynomial interpolation problem, Comp.Sci. Tech.Rep. #942, June, 1990.

[6] MathWorks, MATLAB User's Guide, MathWorks Inc., South Natick MA, February 1989