

**COMPUTATIONAL ASPECTS OF
POLYNOMIAL INTERPOLATION #
IN SEVERAL VARIABLES**

by

**Carl de Boor
and
Amos Ron**

**Computer Sciences Technical Report #924
March 1990**

UNIVERSITY OF WISCONSIN-MADISON
COMPUTER SCIENCES DEPARTMENT

Computational aspects of polynomial interpolation in several variables

Carl de Boor^{1,2} & Amos Ron¹

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 $P \rightarrow \mathbb{R}^\Theta : p \mapsto p|_\Theta$ 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 $\Theta \mapsto \Pi_\Theta$, previously introduced in [BR], for which $\langle \Theta, \Pi_\Theta \rangle$ is always correct, and provide an algorithm for the construction 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, 41A63, 41A10; secondary 65D05, 41A30

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

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

$$P \rightarrow \mathbb{R}^\Theta : p \mapsto p|_\Theta$$

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 $p|_\Theta = f|_\Theta$.

Polynomial interpolation in several variables is hampered by the fact that 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 $P = \Pi_k :=$ the space of all polynomials of degree $\leq k$, 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 [BR], 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 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 show that our algorithm is related to Gauss elimination applied to the Vandermonde matrix (ϑ^α) for Θ , but with a twist. We present a modified algorithm, and provide algorithmic details and some enticing (so we hope) examples.

For alternative approaches in the literature, see, e.g., their discussion in [B9x].

1. The interpolant and some of its properties

The construction proposed in [BR] makes use of the **initial form** 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

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

for f in which $f^{(j)}$ is the sum of all the terms of order j . We also use the abbreviation

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

for any linear space H of functions analytic at the origin.

The initial form or least term is analogous to the **leading** term p_{\uparrow} of a polynomial p in that p_{\uparrow} is the homogeneous polynomial for which $\deg(p - p_{\uparrow}) < \deg p$.

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

$$(1.1) \quad \langle g, f \rangle := \sum_{\alpha} D^{\alpha} g(0) D^{\alpha} f(0) / \alpha!$$

defined, e.g., for an arbitrary function g analytic at the origin and an arbitrary polynomial f . The weights in (1.1) are chosen so that point-evaluation at ϑ is represented with respect to this pairing by the exponential e_{ϑ} , i.e.,

$$(1.2) \quad \langle e_{\vartheta}, f \rangle = f(\vartheta), \quad \vartheta \in \mathbb{R}^s, f \in \Pi,$$

with

$$e_{\vartheta} : \mathbb{R}^d \rightarrow \mathbb{R} : x \mapsto e^{\vartheta \cdot x}$$

the **exponential with frequency** ϑ (and with $\vartheta \cdot x := \sum_j \vartheta(j)x(j)$ the ordinary scalar product). We use the abbreviation

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

for the linear space spanned by the exponentials with frequencies Θ .

In these terms, our assignment for Π_{Θ} is

$$(1.3) \quad \Pi_{\Theta} := (\exp_{\Theta})_{\downarrow}.$$

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

$$(1.4) \quad I_{\Theta} f := \sum_j g_{j\downarrow} \frac{\langle g_j, f \rangle}{\langle g_j, g_{j\downarrow} \rangle},$$

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

$$(1.5) \quad \langle g_i, g_{j\downarrow} \rangle = 0 \quad \iff \quad i \neq j.$$

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 (1.5), it follows that $I_{\Theta} f$ is well-defined and that

$$(1.6) \quad \langle g_i, I_{\Theta} f \rangle = \langle g_i, f \rangle, \quad \text{all } i.$$

But this implies that

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

since g_1, g_2, \dots, g_n is a basis for \exp_Θ .

This also implies that the space

$$\text{span}\{g_{1\downarrow}, \dots, g_{n\downarrow}\}$$

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 [BR]), we must have

$$\Pi_\Theta = \text{span}\{g_{1\downarrow}, \dots, g_{n\downarrow}\}.$$

It may not be clear, offhand, just what the interpolation formula (1.4) may mean in case f is not a polynomial, since we explained the expression $\langle g, f \rangle$ only for polynomial f . The argument just given suggests the answer: Take the numbers $\langle g_j, f \rangle, j = 1, \dots, n$, to be the unique solution to the linear system

$$\sum_i A(\vartheta, i) \langle g_i, f \rangle = f(\vartheta), \quad \vartheta \in \Theta,$$

with the (necessarily invertible) matrix A defined by

$$\sum_i A(\vartheta, i) g_i := e_\vartheta.$$

With the numbers $\langle g_j, f \rangle$ so determined, the resulting $I_\Theta f$ matches f at Θ , whatever the nature of f may be.

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 \implies 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,

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

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. Hence, altogether

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

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

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

(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 , $\dim P \cap \Pi_j \leq \dim \Pi_\Theta \cap \Pi_j$. 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 ‘shadow’ subset of a rectangular grid of points.

(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., 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 (1.3). For example, if $\#\Theta = 1$, then necessarily $\Pi_\Theta = \Pi_0$ (by (7)). If $\#\Theta = 2$, then, by (6) and (7), necessarily $\Pi_\Theta = \Pi_1(\text{affine}(\Theta))$. If $\#\Theta = 3$, then $\Pi_\Theta = \Pi_k(\text{affine}(\Theta))$, with $k := 3 - \dim \text{affine}(\Theta)$. The case $\#\Theta = 4$ is the first one that is not clear-cut. In this case, we have again

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

but only for $k = 1, 3$. When $\text{affine}(\Theta)$ 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 $\Pi_\Theta = \Pi_1 + \text{span}\{q\}$ for some homogeneous quadratic polynomial q . While (2) and (6) impose further restrictions, it seems possible to construct a suitable map $\mathbb{R}^2 \rightarrow \Pi_2^0 : \theta \mapsto q$ 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 $(\Theta, (\text{exp}_\Theta)_1)$ is always correct, and this started us off studying the assignment $\Pi_\Theta := (\text{exp}_\Theta)_1$.

2. The choice of P provided by elimination

In this section, we provide further insight into our particular assignment $\Pi_\Theta = (\text{exp}_\Theta)_1$ 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

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

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 $c := (c(\alpha))_{\alpha \in \mathbb{Z}_+^d}$ solves the linear system

$$(2.1) \quad V? = f|_\Theta,$$

with

$$(2.2) \quad V := \left(\vartheta^\alpha \right)_{\vartheta \in \Theta, \alpha \in \mathbb{Z}_+^d}$$

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

We note that V is of full rank. Perhaps the fastest way to see this is to observe that $(a(\vartheta))_{\vartheta \in \Theta} V = 0$ implies that $\sum_{\vartheta \in \Theta} e_\vartheta a(\vartheta) = 0$ (since $\vartheta^\alpha = (D^\alpha e_\vartheta)(0)$), and thus to rely on the linear independence of any collection of exponentials with distinct frequencies. (Such independence is demonstrated by induction since it is obvious when $\#\Theta = 1$: If $\#\Theta > 1$ and $s := \sum_{\vartheta \in \Theta} e_\vartheta a(\vartheta) = 0$ with all $a(\vartheta) \neq 0$, then also $D_y s = \sum_{\vartheta \in \Theta} (y \cdot \vartheta) e_\vartheta a(\vartheta) = 0$ for any particular y . Since the ϑ are distinct, we can choose y so that $y \perp \theta$ for a particular $\theta \in \Theta$ while $y \not\perp \vartheta$ for at least one $\vartheta \in \Theta$. Thus $\sum_{\vartheta \neq \theta} (y \cdot \vartheta) e_\vartheta a(\vartheta) = 0$ 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.)

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 $\vartheta \in \Theta$.

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), 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 $\beta_1, \beta_2, \dots, \beta_n$ 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 $\{\vartheta_1, \vartheta_2, \dots, \vartheta_n\}$ 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

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

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

$$(2.3) \quad a := (LU)^{-1} (f(\vartheta_1), \dots, f(\vartheta_n))$$

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

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

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

$$\alpha < \beta \quad \implies \quad |\alpha| \leq |\beta|,$$

with

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

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). But P 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

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

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

This simple example also illustrates that the sequence $\beta_1, \beta_2, \dots, \beta_n$ 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 $(())^{\beta_j}$ for the polynomial space P selected, when considered as a space of functions on Θ .

The assignment $\Theta \mapsto \Pi_\Theta$ proposed in [BR] turns out to differ from the above assignment only in 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 $g_i := \sum_\alpha W(\vartheta_i, \alpha)()^\alpha / \alpha!$. Since $\vartheta^\alpha = (D^\alpha e_\vartheta)(0)$, each g_i is in exp_Θ , and since W is in row echelon form, the sequence $g_{1\downarrow}, \dots, g_{n\downarrow}$ is linearly independent, hence a basis for $(\text{exp}_\Theta)_\downarrow = \Pi_\Theta$.

But these particular g_1, g_2, \dots, g_n do not in general satisfy (1.5). To obtain a basis g_1, g_2, \dots, g_n for exp_Θ satisfying (1.5), we carry out elimination, not monomial by monomial, but order by order.

3. Elimination by order

We proposed in [BR] a particular algorithm for the construction of the basis g_1, g_2, \dots, g_n for exp_Θ satisfying (1.5) and needed for (1.4). But, with the details of Gauss elimination recalled in the preceding section in mind, it seems more efficient to construct (as already proposed in [B9x]) the g_i by applying Gauss elimination with partial pivoting to the matrix

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

obtained from the Vandermonde $V = (\vartheta^\alpha)$ by treating all entries of a given order 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 $k = 0, 1, 2, \dots$. Correspondingly,

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

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 (1.1), the relevant scalar product is

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

when eliminating in column k of \mathbf{V} .

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, \dots, k_n and some ordering $\{\vartheta_1, \vartheta_2, \dots, \vartheta_n\}$ 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, \dots, 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

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

is upper triangular and invertible. Consequently, with

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

the matrix

$$\left(\langle \mathbf{G}(\vartheta_i, k_j), \mathbf{G}(\vartheta_j, k_j) \rangle_{k_j} \right)_{i,j=1}^n$$

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

```

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

```

in which the j th 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

$$(3.2) \quad \langle a, b \rangle := \sum_k \langle a, b \rangle_k = \sum_\alpha a(\alpha) b(\alpha) / \alpha!$$

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

$$(3.3) \quad \langle G(\vartheta_i, \cdot), G_{k_j}(\vartheta_j, \cdot) \rangle = \delta_{i,j} / U(j, j), \quad i, j = 1, \dots, n,$$

with G_k given by

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

Now note that the weighted scalar product (3.2) for sequences agrees with the pairing (1.1) in the following sense. For g analytic at the origin and $f \in \Pi$,

$$\langle g, f \rangle = \langle (D^\alpha g(0)), (D^\alpha f(0)) \rangle.$$

Thus, with

$$(3.4) \quad g_i := \sum_\alpha (\cdot)^\alpha / \alpha! G(\vartheta_i, \alpha),$$

we have

$$(3.5) \quad g_{i\downarrow} = \sum_{|\alpha|=k_i} (\cdot)^\alpha / \alpha! G(\vartheta_i, \alpha) = \sum_\alpha (\cdot)^\alpha / \alpha! G_{k_i}(\vartheta_i, \alpha),$$

hence

$$(3.6) \quad \langle g_i, g_{j\downarrow} \rangle = \delta_{i,j} / U(j, j), \quad i, j = 1, \dots, n.$$

Also,

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

since $LUG = V = (D^\alpha e_{\vartheta_i}(0))_{i,\alpha}$. Since g_1, g_2, \dots, g_n is linearly independent by (3.6), this implies that g_1, g_2, \dots, g_n is a basis for \exp_Θ . But (3.6) also implies that $g_{1\downarrow}, \dots, g_{n\downarrow}$ so constructed is linearly independent, hence a basis for Π_Θ .

This proves

(3.7) Proposition. *The functions g_i defined by (3.4) provide a basis for \exp_Θ which satisfies (1.5).*

It follows that

$$I_\Theta f = \sum_i g_{i\downarrow} a(i),$$

with the coefficient vector a obtained as

$$(3.8) \quad a := \text{diag}(U)(LU)^{-1} (f(\vartheta_1), \dots, f(\vartheta_n)),$$

i.e., by a possible permutation of the given vector $f|_{\Theta}$, followed by forward- and backsubstitution. The multiplication by the diagonal matrix $\text{diag}(U)$ accounts for the division by $\langle g_j, g_{j\downarrow} \rangle$ in (1.4), as the latter number is $1/U(j, j)$, by (3.6).

It is worth noting that the factoring out of U will not change the pivot entries since $U(i, j) = 0$ if $k_i = k_j$ and $i \neq j$, except for the normalizing division. This implies that

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

(with W_k defined entirely analogously to G_k), hence 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, \dots, g_n from $f_j := e_{\vartheta_j}$, $j = 1, \dots, n$ proposed in [BR] amounts to Gauss elimination with **column** pivoting applied to \mathbf{V} , except that no columns are actually interchanged. Rather, at the j th step, one looks for the left-most nonzero entry in the j th row of the working array \mathbf{W} , say the entry $\mathbf{W}(\vartheta_j, k_j)$, then uses the j th 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, \dots, k_n ,

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

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

$$g_j := \sum_{\alpha} \binom{\alpha}{\alpha!} W(\vartheta_j, \alpha)$$

satisfy (1.5), 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 [BR] that the two sequences $g_{1\downarrow}, \dots, g_{n\downarrow}$ produced by the two algorithms span the same space.

The algorithm outlined in this section seems preferable to the one from [BR] 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 of 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., [B]).

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**

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

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**): Start with

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

and generate from this

$$(4.2) \quad c(\alpha) := D^{\alpha} p(0) / |\alpha|! + \sum_{i=1}^d x_i c(\alpha + \mathbf{i}_i), \quad |\alpha| = k,$$

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

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

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

(ii) The information about g_j computed by the algorithm outlined in the preceding section readily provides the numbers $D^\alpha g_j(0)$, hence the calculation of the modified power form for $I_\Theta f$ from the matrix G and the vector $(LU)^{-1} f|_\Theta$ does not require generation and use of the multinomial coefficients.

The close similarity to de Casteljau's algorithm (see, e.g., [B]) 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 $\leq k$ 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

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

for $j = k-1, k-2, \dots, 0$, 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

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

for $j = k-1, k-2, \dots, 0$ 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 orders.

5. Algorithmic details

We give here a MATLAB-like program (see, e.g., [M] for language details) for the construction of our interpolant in order to document the simplicity of the actual calculations needed.

```
% INPUT:  $\Theta = \{\vartheta_1, \dots, \vartheta_n\}$ ,  $f|_\Theta$ , 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  $\Theta$ 
    W(:,k) <--  $L^{-1}*V(:,k)$ 
  end
  if i>j, interchange rows i and j
  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  $f|_{\mathcal{e}}$ 
a <-- diag(U)* $U^{-1}*L^{-1}*f$ 
kmax <-- max(K); dk <--  $\binom{[0:kmax]+d-1}{d-1}$ 
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 $c(\alpha) = \text{coefs}(|\alpha|, \alpha)$ for the modified power form of the interpolant, provided that the (scaled) scalar product

$$(5.1) \quad \langle a, b \rangle := \sum_{|\alpha|=k} a(\alpha)b(\alpha) \binom{|\alpha|}{\alpha}$$

is used when working on column k . The needed weights

$$w(\alpha) := \binom{|\alpha|}{\alpha}$$

are integers and are conveniently generated from their recurrence relation

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

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

$$w(0) = 1$$

and the side conditions

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

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

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

and (iii) the use of (4.2).

The output does depend on the choice of the tolerance `tol`. In exact arithmetic, we could choose `tol` = 0. Further, we can be sure that we stay in the `while`-loop at most one time, i.e., that we can find a pivot in column $k+1$ when we run out of pivots in column k . This is due to the fact that Π_Θ is D -invariant (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 j th unit vector in \mathbb{R}^2 . With this ordering of the points, and the lexicographic ordering for the α , the Vandermonde matrix becomes

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

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

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

Hence

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

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 $v \rightarrow 0$, $g_{4\downarrow}$ 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 $(u, v) \rightarrow 0$ or $(u, v) \rightarrow (1, 0)$, 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. On the other hand, any five of these six points are generic, i.e., they are not collinear. Hence

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

with the orthogonal complement $(\Pi_2 \ominus \text{span}(p_\uparrow))$ of $\text{span}(p_\uparrow)$ in Π_2 taken in the sense of the pairing (1.1), and with q a certain homogeneous third-degree polynomial.

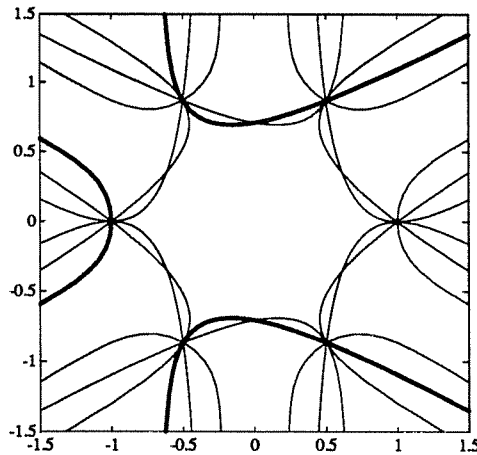
Here is one way to determine this q : Consider the interpolant $I_\Theta f$ to data $f(\vartheta_j) = (-)^j$, all j . There are three lines through the origin not containing any of the interpolation points but such that reflection across that line leaves Θ unchanged. By (6), such reflection must also 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 order 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

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

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 $\Pi_5(\mathbf{C})$, therefore the interpolant).

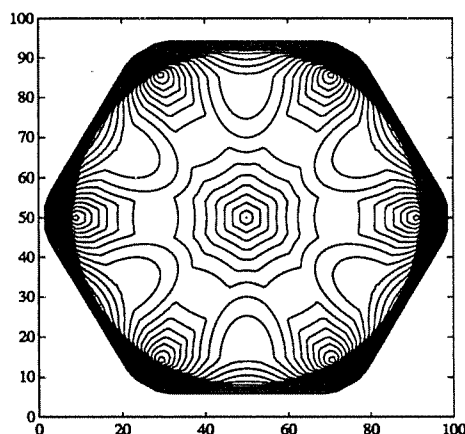


(6.3)Figure. The Lebesgue function for interpolation at the hexagon points is one on the entire central portion.

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 $(\cdot)^{2,0} + (\cdot)^{0,2}$. In particular, for *any* six-pointset Θ on a circle, the homogeneous cubic polynomial in Π_Θ is a linear combination of $\operatorname{Re} z^3$ and $\operatorname{Im} z^3$.

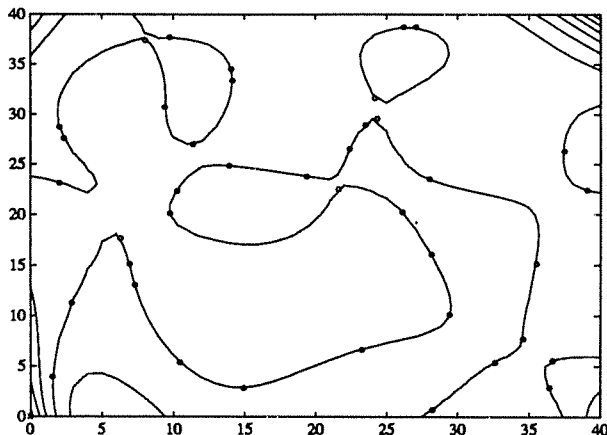
The three heavy lines in (6.3)Figure show the zeros of the (real) Lagrange polynomial ℓ_6 for our interpolation at the hexagon points. This means that ℓ_6 is the unique polynomial in Π_Θ which is 1 at $(1,0)$ and 0 at the other five points. From the figure (and the fact that $\ell_6(1,0) = 1$), we conclude that ℓ_6 is positive near 0. Since Π_Θ is invariant under rotation by $\pi/3$, the remaining five Lagrange polynomials ℓ_j , $j = 1, \dots, 5$, 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 that, *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.*



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

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 $p = 1 - (\cdot)^{2,0} - (\cdot)^{0,2}$ 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. (6.4)Figure shows the level lines of the Lebesgue function, for the values 1:0.05:2. This shows that, on the unit circle, the Lebesgue function does not exceed 1.5.

(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



(6.5)Figure. Contour lines for error in interpolation at 40 random points

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. Proofs

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 [BR]. But some of the consequences mentioned still require proof.

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

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

from (6). 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 $s+1, \dots, d$. This implies that Π_Θ must contain any polynomial p which does not depend on its arguments $s+1, \dots, d$ 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 , $\deg I_\Theta p \leq \deg p$.

(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 $\deg I p \leq \deg p$ for all $p \in \Pi$.*

Proof. Assume that $\langle \Theta, P \rangle$ is correct and that $p \in \Pi$. Extend $I p$ 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 $I p = p$ on Θ , it follows that $(B \setminus I p) \cup p$ is also linearly independent over Θ , hence its span, Q say, is also correct for interpolation on Θ . If now $\deg I p > \deg p =: j$, then it follows that $\dim Q \cap \Pi_j > \dim P \cap \Pi_j$, hence P is not of minimal degree.

Conversely, if $\deg p \geq \deg I p$ 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 $I(\Pi_k \cap Q) \subseteq \Pi_k \cap P$. 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.

Inductive application of (9) shows that

$$(7.2) \quad \Pi_\Theta = \bigotimes_{i=1}^d \Pi_{\langle \# \Theta_i \rangle}(\mathbb{R}) \quad \text{for } \Theta = \times_{i=1}^d \Theta_i.$$

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

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

(in which we take Θ_i to consist of $\gamma(i)$ real points). Here, we call Θ **order-closed** (or, a ‘lower set’) if it is of the form $\Theta = \{\theta_\alpha : \alpha \in \Gamma\}$ for $\theta_\alpha := (x_1(\alpha(1)), \dots, x_d(\alpha(d)))$, with Γ an order-closed subset of C_γ , i.e., $\alpha \in \Gamma$ and $\beta \leq \alpha$ implies $\beta \in \Gamma$, and

$$C_\gamma := \{1, \dots, \gamma(1)\} \times \dots \times \{1, \dots, \gamma(d)\}.$$

Thus,

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

We show now that the standard assignment for this Θ , i.e., the space $\text{span}\{()^\beta : \beta \in \Gamma\}$, 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 $\Pi_\alpha := \text{span}\{()^\beta : \beta \leq \alpha\}$, by (7.2). By (8), each such Π_α must be contained in Π_Θ , hence $\text{span}\{()^\beta : \beta \in \Gamma\} \subset \Pi_\Theta$, 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

$x \ x$	$x \ x \ x \ x \ x \ x$	$x \ x \ x \ x \ x \ x$
$x \ x$	$x \ \ \ \ \ \ x$	$x \ x \ x \ x \ x \ x$
$x \ x$	$x \ \ \ \ \ \ x$	$\ \ \ \ \ \ x \ x$
$x \ x$	$x \ \ \ \ \ \ x$	$\ \ \ \ \ \ x \ x$
$x \ x \ x \ x \ x \ x$	$x \ \ \ \ \ \ x$	$\ \ \ \ \ \ x \ x$
$x \ x \ x \ x \ x \ x$	$x \ x \ x \ x \ x \ x$	$\ \ \ \ \ \ x \ x$

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 [BRxx].

8. Generalizations

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

$$[\vartheta] : f \mapsto f(\vartheta)$$

with ϑ in some n -set Θ .

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

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

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

$$f_i(x) = \lambda_i(\exp^x).$$

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

$$(8.1) \quad \sum_j g_{j\downarrow} \frac{\langle g_j, f \rangle}{\langle g_j, g_{j\downarrow} \rangle}$$

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

This general setting is discussed in [BRxx] in detail, and also for various specific choices of the ‘interpolation conditions’ $\lambda_1, \lambda_2, \dots, \lambda_n$. 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, \dots, f_n for some linear subspace H of A_0 , the homogeneous polynomials $g_{1\downarrow}, \dots, g_{n\downarrow}$ provided by the algorithm form a basis for $H_\downarrow = \text{span}\{f_\downarrow : f \in H\}$. 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 [BR9x].

References

- [B] C. de Boor, B-form basics, in *Geometric Modeling*, G. Farin ed., SIAM (1987), 131-148.
- [B9x] 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.
- [BR] C. de Boor and A. Ron, Multivariate polynomial interpolation, *Constructive Approximation* **x** (199x), xxx-xxx.
- [BR9x] 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.
- [BRxx] C. de Boor and A. Ron, The least solution for the polynomial interpolation problem, ms.
- [M] MathWorks, MATLAB User’s Guide, MathWorks Inc., South Natick MA, February 1989