SIAM J. SCI. COMPUT.
Vol. 22, No. 5, pp. 1717–1740

FAST SOLUTION OF THE RADIAL BASIS FUNCTION INTERPOLATION EQUATIONS: DOMAIN DECOMPOSITION METHODS*

R. K. BEATSON†, W. A. LIGHT‡, AND S. BILLINGS§

Abstract. In this paper we consider domain decomposition methods for solving the radial basis function interpolation equations. There are three interwoven threads to the paper. The first thread provides good ways of setting up and solving small- to medium-sized radial basis function interpolation problems. These may occur as subproblems in a domain decomposition solution of a larger interpolation problem. The usual formulation of such a problem can suffer from an unfortunate scale dependence not intrinsic in the problem itself. This scale dependence occurs, for instance, when fitting polyharmonic splines in even dimensions. We present and analyze an alternative formulation, available for all strictly conditionally positive definite basic functions, which does not suffer from this drawback, at least for the very important example previously mentioned. This formulation changes the problem into one involving a strictly positive definite symmetric system, which can be easily and efficiently solved by Cholesky factorization. The second section considers a natural domain decomposition method for the interpolation equations and views it as an instance of von Neumann's alternating projection algorithm. Here the underlying Hilbert space is the reproducing kernel Hilbert space induced by the strictly conditionally positive definite basic function. We show that the domain decomposition method presented converges linearly under very weak nondegeneracy conditions on the possibly overlapping subdomains. The last section presents some algorithmic details and numerical results of a domain decomposition interpolatory code for polyharmonic splines in 2 and 3 dimensions. This code has solved problems with 5 million centers and can fit splines with 10,000 centers in approximately 7 seconds on very modest hardware.

Key words. radial basis functions, interpolation equations, fast solution method

AMS subject classifications. 65D05, 65D07, 65F10

PII. S1064827599361771

1. Introduction.

One of the most basic problems in approximation theory is to construct an approximation to data specified at m distinct points $x_1, \dots, x_m$ in $\mathcal{R}^n$. A simple approach consists of choosing m functions and then looking for the unique combination of these functions which interpolates the data at the given points. For this process to be successful, the set of functions chosen must be linearly independent over the set of interpolation points $x_1, \dots, x_m$. If the functions are $u_1, \dots, u_m$, then this is equivalent to the matrix $\big(u_j(x_i)\big)$ being invertible. To implement this program, the user must make a choice of the functions $u_1, \dots, u_m$. However, it is sometimes more natural to specify not the actual basis to be used, but rather the subspace given by $\operatorname{span}\{u_1, \dots, u_m\}$. This is the approach we intend to study. We consider interpolants of the form

$$s = p + \sum_{j=1}^{m} \lambda_j \Phi(\cdot, x_j). \tag{1.1}$$

*Received by the editors September 30, 1999; accepted for publication (in revised form) June 14, 2000; published electronically December 28, 2000.
http://www.siam.org/journals/sisc/22-5/36177.html

†Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand (r.beatson@math.canterbury.ac.nz). The work of this author was partially supported by PGSF Subcontract DRF 601 and EPSRC Visiting Fellowship GR/M/52595.

‡Department of Mathematics and Computer Science, University of Leicester, Leicester, LE1 7RH, United Kingdom (pwl@mcs.le.ac.uk). The work of this author was supported by EPSRC grant GR/L/36222.

§Geophysical Technology Ltd., P.O. Box U9, The University of New England, Armidale, NSW 2351, Australia (stephenbillings@yahoo.com).

Here $X = \{x_1, \dots, x_m\}$ is a given set of distinct centers, p is a polynomial of total degree at most $k-1$, and Φ is a mapping from $\mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$. Because there are more parameters than data, we impose the further conditions that the coefficient vector λ satisfies

$$\sum_{j=1}^{m} \lambda_j q(x_j) = 0 \tag{1.2}$$

for all polynomials q of degree at most $k-1$. The related interpolation problem follows.

Problem 1.1. *Given distinct nodes $X = \{x_1, \dots, x_m\} \subset \mathcal{R}^n$ unisolvent with respect to the polynomials of degree at most $k-1$, and a vector $d \in \mathcal{R}^m$, find a function s of the form specified by* (1.1) *and* (1.2)*, satisfying the interpolation conditions*

$$s(x_i) = d_i, \qquad 1 \le i \le m.$$

Let π_{k-1} denote the subspace of $C(\mathcal{R}^n)$ consisting of all polynomials of total degree $k-1$, and let $\{p_1, p_2, \dots, p_\ell\}$ be a basis for π_{k-1}. Then the corresponding system of equations is often written in matrix terms as

$$\begin{pmatrix} \mathsf{A} & \mathsf{P} \\ \mathsf{P}^T & \mathsf{O} \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}, \tag{1.3}$$

in which

$$\mathsf{A}_{ij} = \Phi(x_i, x_j), \qquad i, j = 1, \dots, m, \tag{1.4}$$

and

$$\mathsf{P}_{ij} = p_j(x_i), \qquad i = 1, \dots, m, \quad j = 1, \dots, \ell. \tag{1.5}$$

Note that in this system, and throughout the paper, we use the symbol O to represent the zero matrix of appropriate size, while the symbol 0 is used for the zero vector/scalar.

The above setting covers natural spline interpolation in $\mathcal{R}$ and interpolation by radial basis functions in $\mathcal{R}^n$. In the latter case, the function Φ has the form $\Phi(x, y) = \phi(|x-y|)$ for $x, y \in \mathcal{R}^n$, where $\phi : \mathcal{R} \to \mathcal{R}$, and $|\cdot|$ is the Euclidean norm in $\mathcal{R}^n$. This case is of considerable practical importance. Problem 1.1 is known to have a unique solution for many choices of Φ. To discuss these results we will need the following definition.

Definition 1. *A symmetric function $\Phi : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$ is strictly conditionally positive definite of order k on $\mathcal{R}^n$ if for all sets $X = \{x_1, \dots, x_m\} \subset \mathcal{R}^n$ of distinct points, and all vectors $\lambda \in \mathcal{R}^m$ satisfying the orthogonality condition* (1.2)*, the quadratic form*

$$\lambda^T \mathsf{A} \lambda = \sum_{i,j=1}^{m} \lambda_i \lambda_j \Phi(x_i, x_j)$$

is positive whenever $\lambda \neq 0$.

The paper of Micchelli [11] showed that the interpolation system (1.3) is invertible whenever the function $\Phi(x, y) = \phi(|x-y|)$ is strictly conditionally positive definite of order k, and X is unisolvent for π_{k-1}.[1] Earlier work in a similar vein can be found in Schoenberg [21]. It is worth noting that if a function Φ is conditionally positive definite of order k, then it is also conditionally positive definite of order k' for any $k' \geq k$, and so there is some flexibility over the choice of the parameter k.

For radial basis function interpolation, the most popular choices of the function Φ are strictly conditionally positive definite, notably the multiquadrics and the polyharmonic splines. These interpolation methods are popular for a number of reasons. First, the interpolant depends only on the radial distance between the interpolation points. This feature makes radial basis function interpolation suitable for treating unstructured data—particular data which does not fall on, or close to, the points of some rectangular or regular triangular grid. Such data is often described as scattered. We have already indicated that in practical cases, the types of geometry for which the interpolant is not specified uniquely are extremely simple. For example, interpolation by thin-plate splines in $\mathcal{R}^2$ requires only that the nodes must not lie on a line. With a scattered data set, one can almost guarantee that this condition will be satisfied. This is in sharp contrast to the situation for polynomial interpolation, where the conditions on the data sets required in order that a unique interpolant exists get progressively more complex as the number of interpolation points increases. Furthermore, all the radial basis function interpolants in use have the feature that they minimize a seminorm. Sometimes (as is the case with polyharmonic spline interpolants—see below) the seminorm can be seen to force the interpolant to be the "smoothest" interpolant in some sense. This smoothness property can actually be seen in graphical representations of the interpolant and is akin to the smoothness experienced when interpolating by a low degree polynomial spline in $\mathcal{R}$.

Despite the many nice features of radial basis function interpolation, early workers in the field encountered significant computational difficulties. A small part of these difficulties was due to the computing equipment available at the time, but the major problem was that system (1.3) led to an very ill-conditioned system of equations with a full matrix. We refer the reader to [1] for quotes concerning these difficulties. Further discussion of the ill-conditioning can be found in [10]. In [12] lower bounds on the condition number were given, which showed the poor conditioning of the matrix of system (1.3) as a function of the minimum separation between the interpolation points. It is clear that direct solution methods are not feasible for large data sets. Therefore, a number of iterative methods have been proposed, of which [1] and [4] are recent examples.

This paper focuses on various aspects of the numerical solution of interpolation problems using interpolants of the type described in (1.1). Our prime objective is to present a method for large problems with many, many nodes. However, we will also need good methods for small subproblems that occur naturally in our solution strategy. Two questions arise immediately. First, is (1.1) the best way of couching the interpolation problem, or are there other bases for the finite-dimensional space

$$\mathcal{S} = \left\{ p + \sum_{j=1}^{m} \lambda_j \Phi(\cdot, x_j) : \textstyle\sum_{j=1}^{m} \lambda_j q(x_j) = 0 \text{ for all } q \in \pi_{k-1} \right\}$$

which are numerically superior? Second, once we have decided on a suitable choice of basis, what solver do we use for the linear system generated by this basis?

[1] A suitably modified statement holds when X is not unisolvent for π_{k-1}.

In section 2 we investigate the choice of a basis. We have a number of objectives in this section. First, we do not want to work with a spanning set for $\mathcal{S}$ which is "overspecified" in the sense that it consists of more functions than the dimension of $\mathcal{S}$. Second, we would like the basis to have some natural link to the problem. Third, many of the problems which come within our remit have a certain natural feature—that of scale independence. We want a basis for $\mathcal{S}$, and certainly a method of solution which possesses this same scale independence. Let us be a bit more precise.

DEFINITION 2. *Consider an interpolation method $L_{X,d}$ defined for some suitable class of finite subsets of points $X \subset \mathcal{R}^n$ and data values d. The interpolation method $L_{X,d}$ will be called scale independent if for all suitable sets of interpolation points X and data values d, the interpolant s fitted at the points X to data values d, and the interpolant s^h fitted at the scaled points hX to the unscaled data values d, are related by $s^h(h\,\cdot) = s$.*

As an example, consider the polyharmonic spline interpolants, corresponding to the strictly conditionally positive definite function of order $k+1$,

$$\Phi(x,y) = (-1)^{k+1}|x-y|^{2k}\ln|x-y| \qquad \text{for } x,y \in \mathcal{R}^n.$$

If n is even, then the interpolant corresponding to strict conditional positive definiteness of order $k' = k + (n/2)$, $p \in \pi_{k'-1}$ and sums of shifts of Φ, minimizes the seminorm

$$|g|^2 = \int_{\mathcal{R}^n} \sum_{|\alpha|=k'} \binom{k'}{\alpha} \left(D^\alpha g\right)^2 dx$$

over all suitably smooth interpolating functions.[2] This condition, and the scaling of derivatives, clearly implies that polyharmonic spline interpolation is scale independent. However, we shall see that the numerical methods commonly used for fitting such splines are not scale independent. This is because the chosen basis, or spanning set, behaves badly with respect to scale. We will propose a basis which behaves much better with respect to scale, and which will be very useful for the direct solution of small- to moderately-sized radial basis function interpolation problems.

One particular application for our basis is the direct solution of subproblems arising in a domain decomposition code for large problems. This is the subject of section 4. We wish to suggest that the domain decomposition iterative method of solving a linear system is an effective strategy for the systems considered here. Section 3 introduces a form of the domain decomposition algorithm. This may be viewed as multiplicative Schwarz, or as the alternating projection algorithm of von Neumann [13]. The main benefits of the latter viewpoint are that there is a lot of information available about the convergence of the alternating algorithm and the fact that in favorable circumstances the rate of convergence is linear. We will establish linear convergence of our domain decomposition method, and give some theoretical estimates of the rate, based on the idea of an angle between two subspaces in a Hilbert space.

The description of the domain decomposition method in section 3 is quite general and abstract. In section 4 we present some details of the functioning domain decomposition code and some numerical results.

We conclude this introduction with some remarks about notation. The sign $|\cdot|$ appears frequently throughout the paper and has a number of different meanings. For

[2]It should be clear to the discerning reader that a number of technicalities have been omitted here. Furthermore, it is not really necessary to impose the restriction that n is even, but we wanted to keep the discussion simple.

example, if $x \in \mathcal{R}^n$, then $|x|$ means the Euclidean norm of x. If α is a nonnegative multi-integer, that is, $\alpha \in \mathcal{Z}_+^n$, then $|\alpha|$ is the "length" of α—the sum of its coordinates. Frequent use will be made of the subspace π_{k-1} of $C(\mathcal{R}^n)$ consisting of all algebraic polynomials of total degree at most $k-1$.

2. Alternative bases for the interpolation problem. If the data values $d_1, \ldots, d_m$ are given on $x_1, \ldots, x_m \in \mathcal{R}^n$, then the radial basis function interpolant we seek is a function drawn from the subspace

$$\mathcal{S} = \Bigg\{ s : s = p + \sum_{j=1}^{m} \lambda_j \phi(|\cdot - x_j|) : \ p \in \pi_{k-1}, \lambda_1, \ldots, \lambda_m \in \mathcal{R} \\ \text{and } \sum_{j=1}^{m} \lambda_j q(x_j) = 0 \text{ for all } q \in \pi_{k-1} \Bigg\}.$$

Because the interpolation problem is well posed when Φ is strictly conditionally positive definite of order k, it must be that this space has dimension m. An alternative approach is therefore to develop a basis consisting of only m functions. The first results in this direction were obtained by Sibson and Stone [23], who used the thin-plate spline radial basic function $\Phi(x,y) = |x-y|^2 \log |x-y|$ together with linear polynomials in $\mathcal{R}^2$. As pointed out in the introduction, this choice of radial basic function is a very popular one that has been very successful in applications. In this section we present an alternative approach which is simple to follow both conceptually and numerically and can be carried out for any of the radial basis functions in common usage. It will turn out that our approach is equivalent to that of Sibson and Stone in the case of thin-plate splines in $\mathcal{R}^2$.

The technical fact on which our approach is based is that every basic function which is strictly conditionally positive definite of some order generates a reproducing kernel Hilbert space. That is, there is a Hilbert space $\mathcal{H}$ consisting of continuous functions on $\mathcal{R}^n$ and such that for each $x \in \mathcal{R}^n$ the mapping $f \mapsto f(x)$ for $f \in \mathcal{H}$ is bounded. It then follows easily from the Riesz representation theorem that there is a unique kernel $K : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$ with the following properties.

(i) $K(x,y) = K(y,x)$ for all $x, y \in \mathcal{R}^n$.

(ii) $K(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{R}^n$.

(iii) K is strictly positive definite. That is, for any choice of points $x_1, \ldots, x_m \in \mathcal{R}^n$ and numbers $\lambda_1, \ldots, \lambda_m \in \mathcal{R}$, $\sum_{i,j=1}^{m} \lambda_i \lambda_j K(x_i, x_j) \geq 0$, with equality only when $\boldsymbol{\lambda} = 0$.

(iv) $f(y) = \langle f, K(\cdot, y) \rangle$ for all $f \in \mathcal{H}$ and $y \in \mathcal{R}^n$.

Some other results about reproducing kernels help us in our analysis. Once Φ is given, and the degree of conditional positive definiteness is known, it is possible to write down the reproducing kernel for $\mathcal{H}$ explicitly. To do this we must first make a choice about how to define $\mathcal{H}$. Suppose Φ is strictly conditionally positive definite of order k on $\mathcal{R}^n$. Let $\dim(\pi_{k-1}) = \ell$, and let $\{x_1, \ldots, x_\ell\} \subset \mathcal{R}^n$ be a set of points which is unisolvent with respect to π_{k-1}. The conditional positive definiteness of Φ allows us to define a seminorm $|\cdot|$, with $\mathcal{H}$ then being the set of all functions for which this seminorm is finite. The seminorm has kernel π_{k-1}. The seminorm is modified into a norm simply by choosing an appropriate norm on π_{k-1} and adding it to the seminorm. We always make the choice

$$\|f\|^2 = \sum_{i=1}^{\ell} \big(f(x_i)\big)^2 + |f|^2 \qquad \text{for } f \text{ in } \mathcal{H}.$$

Let $p_1, \ldots, p_\ell$ be the Lagrange basis for π_{k-1} with respect to the points $x_1, \ldots, x_\ell$. That is, the p_j are chosen such that $p_j(x_i)$ is 1 if $i = j$ and is zero otherwise, $1 \le i, j \le \ell$. Then the reproducing kernel K is given by

$$\begin{aligned} K(x,y) &= \Phi(x,y) - \sum_{j=1}^{\ell} p_j(y)\Phi(x,x_j) - \sum_{i=1}^{\ell} p_i(x)\Phi(x_i,y) \\ &\quad + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} p_i(x)p_j(y)\Phi(x_i,x_j) + \sum_{i=1}^{\ell} p_i(x)p_i(y) \quad \text{for } x,y \in \mathcal{R}^n. \end{aligned} \tag{2.1}$$

Here we have tried to give a very rapid sketch of matters which are peripheral to our concerns in this paper. Full details may be found in [8]. Our reason for giving a little of the detail here is that we need the following observations. First, it is easily calculated from (2.1) that $K(\cdot, x_j) = p_j$ for $j = 1, \ldots, \ell$. Second, because of the form of the induced inner product in $\mathcal{H}$ (see (3.1)), $p_1, \ldots, p_\ell$ are an orthonormal basis for π_{k-1}. This follows from the observation that

$$\langle p_\mu, p_\nu \rangle = \sum_{i=1}^{\ell} p_\nu(x_i) p_\mu(x_i)$$

for $\nu, \mu = 1, \ldots, \ell$. Let $P : C(\mathcal{R}^n) \to \pi_{k-1}$ be defined by $Pf = \sum_{j=1}^{\ell} f(x_j)p_j$ for each $f \in C(\mathcal{R}^n)$. The above remarks then show that

$$Pf = \sum_{j=1}^{\ell} f(x_j)p_j = \sum_{j=1}^{\ell} \langle f, K(\cdot, x_j)\rangle p_j = \sum_{j=1}^{\ell} \langle f, p_j \rangle p_j,$$

and so P is the orthogonal projection from $\mathcal{H}$ onto π_{k-1}. It is now possible to compress (2.1) significantly:

$$K = (I \otimes I - P \otimes I)(I \otimes I - I \otimes P)\Phi + \sum_{j=1}^{\ell} p_j \otimes p_j. \tag{2.2}$$

The operator $P \otimes I + I \otimes P - P \otimes P$ is the Boolean sum $(P \otimes I) \oplus (I \otimes P)$. The radial basis interpolant defined in (1.1) and (1.2) can be expressed in the form $s = \sum_{j=1}^{m} \mu_j K(\cdot, x_j)$ as long as the interpolation points $x_1, \ldots, x_m$ are assumed to contain the set $x_1, \ldots, x_\ell$ which is unisolvent for π_{k-1}. This allows us to immediately achieve our first objective: $\{K(\cdot, x_1), \ldots, K(\cdot, x_m)\}$ is a basis for the subspace $\mathcal{S}$, and the interpolant $s = \sum_{j=1}^{m} \mu_j K(\cdot, x_j)$ is specified by the equations

$$\sum_{j=1}^{m} \mu_j K(x_i, x_j) = f_i, \quad 1 \le i \le m. \tag{2.3}$$

The matrix K with elements

$$\mathsf{K}_{ij} = K(x_i, x_j) \quad \text{for } 1 \le i, j \le m \tag{2.4}$$

is positive definite, and so (2.3) gives a potentially very attractive alternative to (1.3) for solving the interpolation equations. Table 1 compares the conditioning of the matrix K with that of the matrix

$$\mathsf{B} = \begin{pmatrix} \mathsf{A} & \mathsf{P} \\ \mathsf{P}^T & \mathsf{O} \end{pmatrix} \tag{2.5}$$

Table 1

Two-norm condition numbers of various matrices corresponding to different formulations of the thin-plate spline interpolation problem. The model problem is on a uniform grid including corners of the unit square with interpoint spacing h along each edge.

Spacing h	Conventional matrix B	Reproducing kernel matrix K	Homogeneous matrix C
1/8	3.5158(3)	1.8930(4)	7.5838(3)
1/16	3.8938(4)	2.6514(5)	1.1086(5)
1/32	5.1363(5)	4.0007(6)	1.6864(6)
1/64	7.6183(6)	6.2029(7)	2.6264(7)

Table 2

Two-norm condition numbers of various matrices corresponding to different formulations of the thin-plate spline interpolation problem. The model problem is on a uniform 5×5 grid in $[0, \alpha]^2$.

Scale parameter α	Conventional matrix B	Reproducing kernel matrix K	Homogeneous matrix C
0.001	2.4349(8)	8.4635(8)	5.4938(2)
0.01	2.4364(6)	8.4640(6)	5.4938(2)
0.1	2.5179(4)	8.5134(4)	5.4938(2)
1.0	3.6458(2)	1.3660(3)	5.4938(2)
10	1.8742(6)	1.2609(3)	5.4938(2)
100	1.1520(11)	1.1396(5)	5.4938(2)
1000	3.4590(15)	1.1386(7)	5.4938(2)

of the usual system (1.3), in which the monomials are used as a basis for the polynomials. Also included is a comparison with a different formulation of the interpolation problem to be developed later. The comparison is over sets of uniformly distributed points in $[0, 1]^2$, using the radial basis function $\phi(r) = r^2 \log r$ with the order of conditional positive definiteness equal to 2. In this table an entry of the form $d_0.d_1d_2d_3(e)$, with d_0, d_1, d_2, d_3 decimal digits, represents the number $d_0.d_1d_2d_3 \times 10^e$. It can be seen that the conditioning of the matrices corresponding to all three approaches is roughly the same, when the problem is posed on the unit square. However, the spectrum of the matrix K of (2.4) and also of the matrix (2.5) are affected by the scale of the problem. For example, Table 2 shows calculated condition numbers for the matrices B of (2.5), the matrix K of (2.4), and the matrix C of the different formulation to be developed later (see Theorem 2.4). These matrices correspond to different ways of casting the interpolation problem. The model problem considered is thin-plate spline interpolation on a uniform 5×5 grid in $[0, \alpha]^2$ with interpoint spacing of $\alpha/4$. The table shows the effect of varying the scale parameter α.

Recall from the introduction that the problem itself is scale independent, so that the scale dependence displayed in Table 2 is not intrinsic to the problem. Rather, it is a consequence of the approach taken to the solving the interpolation system. Some years ago Newsam [15] pointed out to one of us that in the case of the usual interpolation system (1.3) the problem is that the matrix A and the matrix P (using the monomial basis for π_{k-1}) scale differently. We will see later that a similar phenomena is happening in the reproducing kernel setting.

If the solution function we seek is scale independent, then the supplying of data at $hx_1, \ldots, hx_m$, $h > 0$, instead of $x_1, \ldots, x_m$ ought not to change the numerical difficulty of the fitting task. The ideal situation would be if K were homogeneous, that is, if for all $h > 0$ and $x, y \in \mathcal{R}^n$, there were some number $\gamma > 0$ such that $K(hx, hy) = h^\gamma K(x, y)$. This would have the effect that the eigenvalues of the scaled interpolation problem would be scaled versions of the eigenvalues of the unscaled interpolation matrix. Thus, conditioning and eigenvalue clustering would be unaffected by changes of scale. We investigate this possibility next.

LEMMA 2.1. *Let $x_1, \ldots, x_\ell \in \mathcal{R}^n$ be unisolvent with respect to π_{k-1}. Let $p_1, \ldots, p_\ell$ in π_{k-1} be such that $p_i(x_j)$ is 1 if $i = j$ and is zero otherwise, $1 \le i, j \le \ell$. Define $T : C(\mathcal{R}^n \times \mathcal{R}^n) \to C(\mathcal{R}^n \times \mathcal{R}^n)$ by*

$$(Tg)(x,y) = g(x,y) - \sum_{i=1}^{\ell} g(x_i,y)p_i(x) - \sum_{j=1}^{\ell} g(x,x_j)p_j(y) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} p_i(x)p_j(y)g(x_i,x_j)$$

for $x, y \in \mathcal{R}^n$ and $g \in C(\mathcal{R}^n \times \mathcal{R}^n)$. Then T annihilates any function g of the form $g(x,y) = p(x-y)$ with $p \in \pi_{2k-1}$.

Proof. It will suffice to establish the result in the case $p(x) = p_\alpha(x) = x^\alpha$, where $x \in \mathcal{R}^n$ and $\alpha \in \mathcal{Z}_+^n$ with $|\alpha| < 2k$. The binomial theorem provides constants a_β, $\beta \in \mathcal{Z}_+^n$, $0 \le \beta \le \alpha$ such that

$$p(x-y) = \sum_{0\le\beta\le\alpha} a_\beta x^{\alpha-\beta}y^\beta = \sum_{0\le\beta\le\alpha} a_\beta p_{\alpha-\beta}(x)p_\beta(y), \quad x,y \in \mathcal{R}^n.$$

Define $g_{\alpha\beta}$ by $g_{\alpha\beta}(x,y) = p_{\alpha-\beta}(x)p_\beta(y)$. Then,

$$Tp_\alpha = \sum_{0\le\beta\le\alpha} a_\beta T g_{\alpha\beta}. \tag{2.6}$$

Now recall the operator $P : C(\mathcal{R}^n) \to \pi_{k-1}$ given by $Pf = \sum_{j=1}^{\ell} f(x_j)p_j$. Then

$$\begin{aligned} Tg_{\alpha\beta} &= g_{\alpha\beta} - (Pp_{\alpha-\beta})\,p_\beta - p_{\alpha-\beta}Pp_\beta + Pp_{\alpha-\beta}Pp_\beta \\ &= (p_{\alpha-\beta} - Pp_{\alpha-\beta}) \otimes (p_\beta - Pp_\beta)\,. \end{aligned}$$

Now since $|\alpha| \le 2k-1$, it follows that either $|\beta| \le k-1$ or $|\alpha-\beta| \le k-1$ in (2.6). Because the operator $I - P$ annihilates π_{k-1}, the result follows. $\square$

THEOREM 2.2. *Let the symmetric function $\Phi \in C(\mathcal{R}^n \times \mathcal{R}^n)$ be such that $\Phi(hx, hy) = h^\lambda \Phi(x,y) + q_h(x-y)$ for all $h > 0$ and $x, y \in \mathcal{R}^n$, where $\lambda \in \mathcal{R}$ and $q_h \in \pi_{2k-1}$. Let $x_1, \ldots, x_\ell$ be a unisolvent set of points with respect to π_{k-1}, and let $p_1, \ldots, p_\ell$ be the associated Lagrange basis for π_{k-1}: $p_i(x_j) = \delta_{ij}$, $1 \le i, j \le \ell$, where δ_{ij} is the Kronecker delta. Define the symmetric function $H : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$ by*

$$H(x,y) = \Phi(x,y) - \sum_{i=1}^{\ell} p_i(x)\Phi(x_i,y) - \sum_{j=1}^{\ell} p_j(y)\Phi(x,x_j) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} p_i(x)p_j(y)\Phi(x_i,x_j) \quad \text{for } x,y \in \mathcal{R}^n.$$

For $h > 0$, let H^h be defined in the same way except using points $hx_1, \ldots, hx_\ell$ and the associated Lagrange basis for π_{k-1}, $p_1^h, \ldots, p_\ell^h$. Then $H^h(hx, hy) = h^\lambda H(x,y)$ for all $x, y \in \mathcal{R}^n$.

Proof. From the definition

$$H^h(x,y) = \Phi(x,y) - \sum_{i=1}^{\ell} p_i^h(x)\Phi(hx_i,y) - \sum_{j=1}^{\ell} p_j^h(y)\Phi(x,hx_j) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} p_i^h(x)p_j^h(y)\Phi(hx_i,hx_j).$$

Hence,

$$H^h(hx,hy) = \Phi(hx,hy) - \sum_{i=1}^{\ell} p_i^h(hx)\Phi(hx_i,hy) - \sum_{j=1}^{\ell} p_j^h(hy)\Phi(hx,hx_j) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} p_i^h(hx)p_j^h(hy)\Phi(hx_i,hx_j).$$

Clearly $p_j^h(hx) = p_j(x)$ for all $x \in \mathcal{R}^n$ and $1 \le j \le \ell$. Thus,

$$H^h(hx,hy) = h^\lambda \left\{ \Phi(x,y) - \sum_{i=1}^{\ell} p_i(x)\Phi(x_i,y) - \sum_{j=1}^{\ell} p_j(y)\Phi(x,x_j) + \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} p_i(x)p_j(y)\Phi(x_i,x_j) \right\} + (Tv)(x,y),$$

where $v(x,y) = q_h(x-y)$ for some $q_h \in \pi_{2k-1}$, and T is as in Lemma 2.1. By that lemma, $Tv = 0$ and so $H^h(hx,hy) = h^\lambda H(x,y)$ for all $h > 0$ and $x, y \in \mathcal{R}^n$. ∎

We now give an application of Theorem 2.2 to the popular polyharmonic splines.

COROLLARY 2.3. *The function Φ defined by $\Phi(x,y) = (-1)^{k+1}|x-y|^{2k}\ln|x-y|$ is strictly conditionally positive definite of order k' for any $k' \ge k+1$. The corresponding function H satisfies $H^h(hx,hy) = h^{2k}H(x,y)$ for all $x, y \in \mathcal{R}^n$.*

Proof. The conditional positive definiteness can be found in Micchelli [11]. It follows that the reproducing kernel K, and hence the kernel H, are built using points $x_1, \ldots, x_m \in \mathcal{R}^n$ which are unisolvent with respect to $\pi_{k'-1}$. Furthermore,

$$\begin{aligned}\Phi(hx,hy) = (-1)^{k+1}|hx-hy|^{2k}\ln|hx-hy| &= (-1)^{k+1}h^{2k}|x-y|^{2k}\left(\ln|x-y| + \ln h\right)\\ &= h^{2k}\Phi(x,y) + (-1)^{k+1}h^{2k}|x-y|^{2k}\ln h\\ &= h^{2k}\Phi(x,y) + q_h(x-y),\end{aligned}$$

where $q_h \in \pi_{2k} \subset \pi_{2k'-1}$. Theorem 2.2 now applies and $H^h(hx,hy) = h^{2k}H(x,y)$. ∎

The problem with the reproducing kernel matrix is now clearly highlighted. We can write

$$K^h(x,y) = H^h(x,y) + \sum_{i=1}^{\ell} p_i^h(x)p_i^h(y) \quad \text{for } x, y \in \mathcal{R}^n. \tag{2.7}$$

In many cases the H^h function scales at a rate controlled by the basis function with which it is associated. As we just saw in Corollary 2.3, in the thin-plate case with

$\phi(r) = (-1)^{k+1} r^{2k} \ln r$, the kernel H^h scales as a function of h^{2k}. The last term $\sum_{i=1}^{\ell} p_i^h(x) p_i^h(y)$ does not scale at all. Consequently, the general position is that the reproducing kernel is made up of two pieces which scale in fundamentally different ways.

An alternative approach consists of trying to use $\{H(\cdot, x_j)\}_{j=1}^m$ as a basis for $\mathcal{S}$. This is, of course, doomed to failure, because we somehow need the terms $\sum_{i=1}^{\ell} p_i^h(x) p_i^h(y)$ which differentiate between K and H. Indeed, one simply checks that $H(\cdot, x_j) = 0$ for $j = 1, \ldots, \ell$. Consequently, we choose as a potential basis the functions $\{p_1, \ldots, p_\ell, H(\cdot, x_{\ell+1}), \ldots, H(\cdot, x_m)\}$. To solve the interpolation problem using this basis we must find $c_1, \ldots, c_\ell, \gamma_{\ell+1}, \ldots, \gamma_m$ such that

$$\sum_{j=1}^{\ell} c_j p_j(x_i) + \sum_{j=\ell+1}^{m} \gamma_j H(x_i, x_j) = d_i, \quad i = 1, \ldots, m.$$

This system of equations has some special structure which we now investigate. First, from the remark above that $H(\cdot, x_j) = 0$ for $1 \le j \le \ell$ and symmetry, it follows in particular that $H(x_i, x_j) = 0$ for $1 \le i \le \ell$ and all j. Thus our system of equations can be written

$$\begin{pmatrix} \mathsf{I} & \mathsf{O} \\ \mathsf{E}^T & \mathsf{C} \end{pmatrix} \begin{pmatrix} c \\ \gamma \end{pmatrix} = d, \tag{2.8}$$

where I is the $\ell \times \ell$ identity, C is the matrix $\big(H(x_i, x_j)\big)_{i,j=\ell+1}^m$, E^T is the $(m-\ell) \times \ell$ matrix $\big(p_j(x_i)\big)_{i=\ell+1, j=1}^{m,\ \ \ell}$, and c, γ, d are the vectors $(c_1, \ldots, c_\ell)^T$, $(\gamma_{\ell+1}, \ldots, \gamma_m)^T$, and $(d_1, \ldots, d_m)^T$, respectively.

THEOREM 2.4. *The matrix $\mathsf{C} = \big(H(x_i, x_j)\big)_{i,j=\ell+1}^m$ is strictly positive definite.*

Proof. As before, define $P : C(\mathcal{R}^n) \to \pi_{k-1}$ by $Pf = \sum_{j=1}^{\ell} f(x_j) p_j$. Then P is a projection onto π_{k-1}. Furthermore, from (2.2),

$$\begin{aligned} PK(\cdot, x_j) &= [(P \otimes I)\, K](\cdot, x_j) \\ &= [(P \otimes I)(I \otimes I - P \otimes I)(I \otimes I - I \otimes P)\, \Phi](\cdot, x_j) + \sum_{i=1}^{\ell} p_i(x_j) P p_i. \end{aligned}$$

Because $P \otimes I$ is a projection, $(P \otimes I)(I \otimes I - P \otimes I) = 0$ and so $PK(\cdot, x_j) = \sum_{i=1}^{\ell} p_i(x_j) p_i$. This calculation shows that for $j = \ell+1, \ldots, m$,

$$H(\cdot, x_j) = K(\cdot, x_j) - \sum_{i=1}^{\ell} p_i(x_j) p_i = K(\cdot, x_j) - PK(\cdot, x_j). \tag{2.9}$$

Now let $v = (v_{\ell+1}, \ldots, v_m) \in \mathcal{R}^{m-\ell}$. Then,

$$\begin{aligned} v^T \mathsf{C} v &= \sum_{i,j=\ell+1}^{m} v_i v_j H(x_i, x_j) \\ &= \sum_{i,j=\ell+1}^{m} v_i v_j \big\langle H(\cdot, x_i), K(\cdot, x_j) \big\rangle \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i,j=\ell+1}^{m} v_i v_j \big\langle H(\cdot,x_i), H(\cdot,x_j) + PK(\cdot,x_j)\big\rangle \\
&= \sum_{i,j=\ell+1}^{m} v_i v_j \left\{ \big\langle H(\cdot,x_i), H(\cdot,x_j)\big\rangle + \big\langle H(\cdot,x_i), PK(\cdot,x_j)\big\rangle \right\} \\
&= \sum_{i,j=\ell+1}^{m} v_i v_j \left\{ \big\langle H(\cdot,x_i), H(\cdot,x_j)\big\rangle + \big\langle K(\cdot,x_i) - PK(\cdot,x_i), PK(\cdot,x_j)\big\rangle \right\} \\
&= \sum_{i,j=\ell+1}^{m} v_i v_j \big\langle H(\cdot,x_i), H(\cdot,x_j)\big\rangle = \left\| \sum_{j=\ell+1}^{m} v_j H(\cdot,x_j) \right\|^2 \geq 0.
\end{aligned}
$$

In the second line we have used the reproducing property of K; in the third line we have used (2.9); and in the last line we have used the fact that $K(\cdot,x_j) - PK(\cdot,x_j)$ is orthogonal to polynomials of degree at most $k-1$. Furthermore, by the linear independence of $\left\{K(\cdot,x_j)\right\}_{j=1}^{m}$ and the fact that $K(\cdot,x_j) = p_j$, $1 \leq j \leq \ell$, the functions $\{H(\cdot,x_j), j=\ell+1,\ldots,m\}$ are linearly independent in the Hilbert space. Hence, $\sum_{j=\ell+1}^{m} v_j H(\cdot,x_j)$ is nonzero unless $v=0$. It follows that $v^T \mathsf{C} v > 0$ unless $v=0$ and so C is strictly positive definite. □

To summarize, we have shown that using a basis of the Lagrange polynomials, $\{p_j\}_{j=1}^{\ell}$, supplemented by the functions $\{H(\cdot,x_j) : \ell+1 \leq j \leq m\}$ casts the interpolation problem, Problem 1.1, in the form of (2.8) in which C is a strictly positive definite, symmetric matrix. One way to solve this system is as follows. First "solve" for the coefficients of the Lagrange polynomials setting $c = (d_1, \ldots, d_\ell)^T$. Next substitute the value for c back into the second row of the partitioned system (2.8), obtaining with $d^* = (d_{\ell+1}, \ldots, d_m)^T$ the equation

$$
\mathsf{C}\gamma = d^* - \mathsf{E}^T c = \begin{pmatrix} -\mathsf{E}^T & \mathsf{I} \end{pmatrix} d.
$$

In this system the right-hand side $\begin{pmatrix} -\mathsf{E}^T & \mathsf{I} \end{pmatrix} d$ is cheap to calculate as each row of E^T has at most ℓ nonzero entries. Once the right-hand side is calculated, the system can be efficiently solved for γ via Cholesky decomposition of C followed by forward and back substitution. This of course takes approximately half the operations of any Gauss elimination routine that does not exploit symmetry, such as might be applied to the usual system (1.3). Also, throughout this whole process one needs to store only E^T and the lower triangle of C, so that the method is efficient in storage as well as operations.

We conclude this section with a brief exploration of the connection between our work and that of Sibson and Stone [23]. Sibson and Stone examined the radial basis function $\phi(r) = r^2 \ln r$ supplemented by linear ploynomials, using data specified at points of $\mathcal{R}^2$. An extension of their approach to general dimension d, and general order k of polynomial, is as follows. Pre- and postmultiply the usual system

$$
\begin{pmatrix} \mathsf{A} & \mathsf{P} \\ \mathsf{P}^T & \mathsf{O} \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix} \tag{2.10}
$$

by an $m \times (m-\ell)$ matrix Q whose columns span the orthogonal complement of the column space of P. This implies in particular that Q has full rank. Then any $\lambda \in \mathcal{R}^m$ satisfying $\mathsf{P}^T\lambda = 0$ can be uniquely expressed in the form $\lambda = \mathsf{Q}\gamma$, where $\gamma \in \mathcal{R}^{m-\ell}$. Thus, the constraints on λ are automatically satisfied, and we need only solve the unconstrained system $\mathsf{A}\mathsf{Q}\gamma + \mathsf{P}c = d$. Premultiplying by Q^T gives

$$(\mathsf{Q}^T\mathsf{A}\mathsf{Q})\gamma = \mathsf{Q}^T d. \tag{2.11}$$

Observe that for any $u \in \mathcal{R}^{m-\ell}$ with $u \neq 0$, $u^T\left(\mathsf{Q}^T\mathsf{A}\mathsf{Q}\right)u = (\mathsf{Q}u)^T\,\mathsf{A}\,(\mathsf{Q}u) > 0$, since Q has full rank and Φ is strictly conditionally positive definite of order k. Hence, $\mathsf{Q}^T\mathsf{A}\mathsf{Q}$ is a strictly positive definite, symmetric matrix. Therefore, (2.11) uniquely specifies γ and $\lambda = \mathsf{Q}\gamma$. Note also that (2.11) can be rewritten as

$$0 = \mathsf{Q}^T\left(d - \mathsf{A}\mathsf{Q}\gamma\right) = \mathsf{Q}^T\left(d - \mathsf{A}\lambda\right).$$

Hence, $d - \mathsf{A}\lambda$ is in the column space of P, and there is a unique c such that

$$\mathsf{P}c = d - \mathsf{A}\lambda.$$

Thus, we may solve the interpolation problem, Problem 1.1, in the following way.

- Choose an $m \times (m-\ell)$ matrix Q whose columns span the orthogonal complement of the column space of P.
- Find γ by solving the $(m-\ell) \times (m-\ell)$ system (2.11), for example, via Cholesky factorization.
- Set $\lambda = \mathsf{Q}\gamma$.
- Find the polynomial from π_{k-1} interpolating to the following residual function

$$r = d - \sum_{j=1}^{m} \lambda_j \Phi(x, x_j),$$

at the nodes $\{x_1, \ldots, x_\ell\}$.

A possible choice for Q is

$$\mathsf{Q} = \begin{pmatrix} -p_1(x_{\ell+1}) & -p_1(x_{\ell+2}) & -p_1(x_{\ell+3}) & \ldots & -p_1(x_m) \\ \vdots & \vdots & \vdots & & \vdots \\ -p_\ell(x_{\ell+1}) & -p_\ell(x_{\ell+2}) & -p_\ell(x_{\ell+3}) & \ldots & -p_\ell(x_m) \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}.$$

This choice of Q is clearly of full rank. Also, we claim that the columns of Q are orthogonal to the columns of P, that is, $\mathsf{P}^T\mathsf{Q} = \mathsf{O}$. Since each row of P^T is of the form $\big(q(x_1), \ldots, q(x_m)\big)$, for some $q \in \pi_{k-1}$, a typical element of $\mathsf{P}^T\mathsf{Q}$ has the form

$$q(x_{\ell+i}) - \sum_{j=1}^{\ell} q(x_j)p_j(x_{\ell+i}) = q(x_{\ell+i}) - q(x_{\ell+i}) = 0,$$

by the properties of the Lagrange basis.

For the particular choice of Q given above, the matrix C in (2.8) obtained via homogenizing the reproducing kernel turns out to be identical with the matrix $\mathsf{Q}^T\mathsf{A}\mathsf{Q}$ obtained via the generalized Sibson and Stone approach.

Powell [16], in the setting of thin-plate splines in $\mathcal{R}^2$, and building on the Sibson and Stone approach, proposes a different choice of Q. His choice is based upon a Householder QR factorization of P.

Our choice of Q reduces the operation count for the setup of the lower triangle of the $\mathsf{Q}^T\mathsf{A}\mathsf{Q}$, and the modified right-hand side $\mathsf{Q}^T d$. When the iterative solution of the large system requires solution of a large number of different small systems, the accumulated setup cost can be a considerable part of the overall cost of a solve. A further convenience of our choice of Q is that it does not intermix the terms $\Phi(\cdot, x_j)$ as much. In certain situations this can allow one to curtail Cholesky back substitutions early, thus saving additional operations.

3. Convergence and the alternating algorithm. This section discusses the convergence of a domain decomposition method for solving the interpolation equations. In the language of the domain decomposition literature the method analyzed is a multiplicative Schwarz type method. It can be viewed equally well as an instance of the von Neumann alternating projection algorithm. We will show that the method presented converges linearly under very weak assumptions on the choice of the subdomains.

The key to the analysis is to view the algorithm as a sequence of orthogonal projections onto overlapping subspaces. Such an approach is standard in the domain decomposition literature [24, 2] and in the analysis of the von Neumann alternating algorithm [13, 14, 25]. In the radial basis function literature, related Hilbert space ideas are used to advantage by Madych and Nelson [9], Faul and Powell [4], Schaback [18, 19] and Schaback and Wendland [20]. After completing this work we learned of some work on different projection methods by Wenz [26].

As was already emphasized in the previous section, interpolation problems involving translates of a basic function can be solved with the aid of the reproducing kernel $K : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}$. It has been known for some time now that this reproducing kernel can be used to define an inner product on a suitable space of continuous functions. It is not necessary for us to operate at the full level of generality of this result. We can simply suppose that f and g mapping $\mathcal{R}^n \to \mathcal{R}$ have the form

$$f = \sum_{i=1}^{m} a_i K(\cdot, x_i) \quad \text{and} \quad g = \sum_{j=1}^{m} b_j K(\cdot, x_j).$$

Define

$$\langle f, g \rangle = \sum_{i,j=1}^{m} a_i b_j K(x_i, x_j). \tag{3.1}$$

There are a number of minor observations at this point. First of all, there is no loss of generality in our requirement that f and g are both linear combinations of $K(\cdot, x_1), \ldots, K(\cdot, x_m)$. If this is not the case, then one simply coalesces the sets on which f and g are based, and sets the appropriate coefficients amongst the a_i and b_j to zero. Second, the symmetry of the reproducing kernel produces the symmetry of the inner product. Third, since K is a reproducing kernel, the matrix $\big(K(x_i, x_j)\big)_{i,j=1}^{m}$ is positive definite. In all cases of interest to us this matrix is, in fact, strictly positive definite,[3] and so $\langle \cdot, \cdot \rangle$ defines an inner product on the set

$$\left\{ \sum_{j=1}^{m} a_j K(\cdot, x_j) : m \in \mathcal{N}, a_1, \ldots, a_m \in \mathcal{R}, x_1, \ldots, x_m \text{ are distinct points of } \mathcal{R}^n \right\}.$$

[3]For some discussion of what can happen in the exceptional cases, see Shapiro [22, p. 71].

Let $\mathcal{H}$ denote the completion of this space with respect to our inner product. In all cases of interest to us, $\mathcal{H} \subset C(\mathcal{R}^n)$, and so we shall assume this throughout what follows.

Throughout this section we will employ the following notation. Given a finite set of distinct points in $\mathcal{R}^n$, $X = \{x_1, \dots, x_m\}$, we will use the symbol $\mathcal{X}$ to denote the subspace of functions in $\mathcal{H}$ "carried" by X. That is,

$$\mathcal{X} = \left\{ f \in \mathcal{H} : f = \sum_{x \in X} \lambda_x K(\cdot, x), \text{ where } \lambda_x \in \mathcal{R} \right\}.$$

This notation will chiefly be employed with sets of points X or Y, and corresponding sets of functions $\mathcal{X}$ and $\mathcal{Y}$. All these sets will on occasion carry integer subscripts.

Given such a finite set of distinct points $X \subset \mathcal{R}^n$ we consider interpolation problems of the following form:

Given $f \in \mathcal{H}$, find the element of $\mathcal{X}$ that interpolates to f on X.

One version of the domain decomposition method is to let $X_1, X_2, \dots, X_k$ be subsets of X, such that $\cup_{i=1}^k X_i = X$. Given $f \in \mathcal{H}$, we let $P_i f$, $i = 1, \dots, k$, denote the function from $\mathcal{X}_i$ that interpolates to f on X_i. Thus,

$$P_i f = \sum_{x \in X_i} \lambda_x K(\cdot, x) \quad \text{and} \quad (P_i f)(y) = f(y) \quad \text{for all } y \in X_i.$$

The domain decomposition algorithm now generates the sequence of residual vectors $\{f_{nk+r}\}$, where $n \in \mathcal{N}_0$ and $r = 1, 2, \dots, k$ via the relationships

$$f_0 = f \text{ and } f_{nk+r} = f_{nk+r-1} - P_r f_{nk+r-1}.$$

The corresponding sequence of approximations $\{s_{nk+r}\}$ is generated via the relationships

$$s_0 = 0 \text{ and } s_{nk+r} = s_{nk+r-1} + P_r f_{nk+r-1}.$$

The next two results identify interpolation on a subset as the orthogonal projection onto the function space "carried" by that subset. They are key observations utilized by Faul and Powell [4] and Schaback and Wendland [20].

LEMMA 3.1. *Let X be a finite set of distinct points in $\mathcal{R}^n$. Let $f \in \mathcal{H}$ be such that $f(x) = 0$ for all $x \in X$. Let $g \in \mathcal{X}$. Then $\langle f, g \rangle = 0$.*

Proof. Let $X = \{x_1, \dots, x_m\}$ and $g = \sum_{j=1}^m a_j K(\cdot, x_j)$. First suppose that f has the form $f = \sum_{j=1}^n b_j K(\cdot, x_j)$, where $n \geq m$. Set $a_{m+1} = \cdots = a_n = 0$. Then

$$\begin{aligned}
\langle f, g \rangle &= \sum_{i,j=1}^n a_i b_j K(x_i, x_j) \\
&= \sum_{i=1}^m a_i \sum_{j=1}^n b_j K(x_i, x_j) \\
&= \sum_{i=1}^m a_i f(x_i) = 0.
\end{aligned}$$

If $f \in \mathcal{H}$ does not have the form above, then we can find a sequence $\{h_\ell\} \subset \mathcal{H}$ such that each h_ℓ has the form $h_\ell = \sum_{j=1}^{n_\ell} b_{j\ell} K(\cdot, x_j)$ and $\langle f - h_\ell, f - h_\ell \rangle \to 0$ as $\ell \to \infty$. The above calculations show that

$$\langle h_\ell, g \rangle = \sum_{i=1}^{m} a_i h_\ell(x_i).$$

Since the point evaluation functionals lie in $\mathcal{H}^*$ (because $\mathcal{H}$ is a reproducing kernel Hilbert space) it follows that $h_\ell(x_i) \to 0$ as $\ell \to \infty$ for $i = 1, \dots, m$. Hence,

$$\langle f, g \rangle = \lim_{\ell \to \infty} \langle h_\ell, g \rangle = \sum_{i=1}^{m} a_i \lim_{\ell \to \infty} h_\ell(x_i) = 0. \qquad \square$$

COROLLARY 3.2. *Let X be a finite set of distinct points of $\mathcal{R}^n$, and let $\mathcal{X} = \left\{ f \in \mathcal{H} : f = \sum_{x \in X} \lambda_x K(\cdot, x), \text{ where } \lambda_x \in \mathcal{R} \right\}$ be the space "carried" by X. Then*

$$\mathcal{X}^\perp = \{ f \in \mathcal{H} : f(x) = 0 \text{ for all } x \in X \}.$$

Furthermore, let P denote the operator defined by interpolation from $\mathcal{X}$ at the nodes in X. Then P is the orthogonal projection from $\mathcal{H}$ onto $\mathcal{X}$.

Proof. That functions $f \in \mathcal{H}$ zero everywhere on X are in $\mathcal{X}^\perp$ follows directly from Lemma 3.1. In the other direction if f is nonzero at a point $y \in X$, then consider the function $g := K(\cdot, y) \in \mathcal{X}$. By an argument similar to that of the lemma we find $\langle f, g \rangle = f(y) \neq 0$. Hence $f \notin \mathcal{X}^\perp$. This establishes the first claim of the corollary.

Now turn to the second claim. From the interpolation conditions, $f - Pf$ is zero at every point of $x \in X$. Hence, by the lemma $f - Pf$ is orthogonal to $\mathcal{X}$. The second claim now follows from the characterization of the orthogonal projection of $f \in \mathcal{H}$ onto a closed subspace $\mathcal{G}$ of $\mathcal{H}$, as the unique $g \in \mathcal{G}$ such that the error $f - g$ is orthogonal to $\mathcal{G}$. $\square$

It is now possible to understand the domain decomposition algorithm as a version of the alternating projection algorithm of von Neumann [13, 14] or the Kacmarz procedure [6]. Each operator P_i in the domain decomposition algorithm is, in fact, the orthogonal projection from $\mathcal{H}$ onto the subspace

$$\mathcal{X}_i = \left\{ f \in \mathcal{H} : f = \sum_{x \in X_i} \lambda_x K(\cdot, x), \text{ where } \lambda_x \in \mathcal{R} \right\}.$$

If we define $Q_i = I - P_i$, then Q_i is the orthogonal projection onto $\mathcal{X}_i^\perp$, and ℓ complete cycles of the domain decomposition algorithm computes $(Q_k \cdots Q_1)^\ell f$.

The convergence of the alternating algorithm is well understood in terms of the angles between these subspaces $\mathcal{X}_i^\perp$.

DEFINITION 3. *Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be closed subspaces of a Hilbert space $\mathcal{H}$ with $\mathcal{U} = \mathcal{U}_1 \cap \mathcal{U}_2$. Then the angle α between $\mathcal{U}_1$ and $\mathcal{U}_2$ is given by*

$$\cos \alpha = \sup \left\{ \langle u, v \rangle : u \in \mathcal{U}_1 \cap \mathcal{U}^\perp, v \in \mathcal{U}_2 \cap \mathcal{U}^\perp \text{ and } \|u\|, \|v\| \leq 1 \right\}.$$

The following theorem of Smith, Solmon, and Wagner [25] then shows that the alternating algorithm usually converges linearly.[4]

[4]It is well known that the rate estimate of Theorem 3.3 is often pessimistic. However, our interest at this point is to guarantee linear convergence, rather than to obtain the best rate. Rate estimates for the alternating algorithm which are often improvements on the estimate of Theorem 3.3 can be found in [5, 7].

THEOREM 3.3. *Let $Q_1, \ldots, Q_k$ be orthogonal projections onto closed subspaces $\mathcal{U}_1, \ldots, \mathcal{U}_k$ in a Hilbert space $\mathcal{H}$. Let $\mathcal{U} = \cap_{i=1}^k \mathcal{U}_i$. Let $Q : \mathcal{H} \to \mathcal{U}$ be the orthogonal projection onto $\mathcal{U}$, and let α_j be the angle between $\mathcal{U}_j$ and $\mathcal{A}_{j+1} = \cap_{i=j+1}^k \mathcal{U}_j$. Then for any $f \in \mathcal{H}$,*

$$\left\| (Q_k \cdots Q_1)^\ell f - Qf \right\|^2 \le c^{2\ell} \|f - Qf\|^2,$$

where

$$c^2 \le 1 - \Pi_{j=1}^{k-1} \sin^2 \alpha_j.$$

Of course it is possible that the angle between some pair $\mathcal{U}_j$ and A_{j+1} may be zero so that $c = 1$, thus denying us the chance of verifying convergence. However, we will soon see that for our application some very weak conditions on the intersections of the node sets are sufficient to rule out this possibility.

DEFINITION 4. *Let $X_1, \ldots, X_k$ be nonempty subsets of $\mathcal{R}^n$ and let $Y_j = \cup_{i=j}^k X_i$, $j = 1, \ldots k$. The sets $X_1, \ldots, X_k$ will be called* weakly distinct *if for each $j = 1, \ldots, k-1$, $X_j \ne Y_j$ and $Y_{j+1} \ne Y_j$.*

Note that if each X_j has a point which belongs only to it, and to no other X_i, then the sets $X_1, \ldots, X_k$ are weakly distinct.

LEMMA 3.4. *Let $X_1, \ldots, X_k$ be finite subsets of $\mathcal{R}^n$, and let $Y_j = \cup_{i=j}^k X_i$, $j = 1, \ldots, k$. Then,*

(i) $\mathcal{Y}_j^\perp = \cap_{i=j}^k \mathcal{X}_i^\perp$ *for $j = 1, \ldots, k$.*

(ii) *If the sets $X_1, \ldots, X_k$ are weakly distinct, then the subspaces $\mathcal{X}_j^\perp \cap \mathcal{Y}_j$ and $\mathcal{Y}_{j+1}^\perp \cap \mathcal{Y}_j$ contain nonzero functions for $j = 1, \ldots, k-1$.*

Proof. First,

$$\begin{aligned}
\bigcap_{i=j}^k \mathcal{X}_i^\perp &= \bigcap_{i=j}^k \{f \in \mathcal{H} : f(x) = 0 \text{ for all } x \in X_i\} \\
&= \left\{ f \in \mathcal{H} : f(x) = 0 \text{ for all } x \in \bigcup_{i=j}^k X_i \right\} \\
&= \left\{ f \in \mathcal{H} : f = \sum_{x \in Y_j} \theta_x K(\cdot, x) : \theta_x \in \mathcal{R} \right\}^\perp \\
&= \mathcal{Y}_j^\perp.
\end{aligned}$$

This shows the first claim of the lemma.

Since $X_1, \ldots, X_k$ are weakly distinct, there is a point $z \in Y_j$ such that $z \notin X_j$. Now consider an element $u \in \mathcal{Y}_j$ of the form

$$u = \sum_{x \in X_j} \theta_x K(\cdot, x) + K(\cdot, z).$$

We claim there is a choice of the θ_x's such that $u \in \mathcal{X}_j^\perp$. In order that $u \in \mathcal{X}_j^\perp$, we have to satisfy the equations

$$\sum_{x \in X_j} \theta_x K(a, x) = -K(a, z) \text{ for all } a \in X_j.$$

Since the matrix $\big(K(a,x)\big)_{a,x\in X_j}$ is invertible, this set of equations has a unique solution for $\{\theta_x : x \in X_j\}$. The corresponding u is nontrivial since the coefficient of $K(\cdot,z)$ in its expansion is 1. The proof that $\mathcal{Y}_{j+1}^{\perp}\cap\mathcal{Y}_j$ contains a nonzero function is similar and will be omitted. ∎

THEOREM 3.5. *Let $f\in\mathcal{H}$ and let $X_1,\dots,X_k$ be weakly distinct finite subsets of $\mathcal{R}^n$. Set $Y_j=\cup_{i=j}^k X_i$, $j=1,\dots,k$. Let $Q:\mathcal{H}\to\mathcal{Y}_1^{\perp}$ be the orthogonal projection onto $\mathcal{Y}_1^{\perp}$. Let f_ℓ be the element of $\mathcal{H}$ obtained by applying ℓ complete cycles of the domain decomposition algorithm to f on $X_1,\dots,X_k$. Then there exists a number $0\le c<1$ such that*

$$\|f_\ell - Qf\|^2 \le c^{2\ell}\|f-Qf\|^2 \le c^{2\ell}\|f\|^2, \qquad \ell=1,2,\dots.$$

Proof. We know from our earlier discussion that $f_\ell=(Q_k\cdots Q_1)^\ell f$, where $Q_i:\mathcal{H}\to\mathcal{X}_i^{\perp}$ is the orthogonal projection. An application of Theorem 3.3 shows that

$$\left\|(Q_k\cdots Q_1)^\ell f - Qf\right\|^2 \le c^{2\ell}\|f-Qf\|^2, \qquad \ell=1,2,\dots,$$

where c may be estimated in terms of the angles between subspaces.

It remains to show that $c<1$. In the notation of Theorem 3.3, this involves examining α_j, the angle between the subspaces $\mathcal{U}_j=\mathcal{X}_j^{\perp}$ and $\mathcal{A}_{j+1}=\cap_{i=j+1}^k\mathcal{X}_i^{\perp}$. A critical ingredient of this analysis is the orthogonal complement of the intersection of these two spaces, namely the subspace

$$\mathcal{B}_j=\left(\mathcal{X}_j^{\perp}\bigcap\left(\bigcap_{i=j+1}^{k}\mathcal{X}_i^{\perp}\right)\right)^{\perp}=\left(\bigcap_{i=j}^{k}\mathcal{X}_i^{\perp}\right)^{\perp}.$$

In Lemma 3.4 it was shown that $\mathcal{Y}_j^{\perp}=\left(\cap_{i=j}^k\mathcal{X}_i^{\perp}\right)$ for $j=1,\dots,k$. It follows that $\mathcal{A}_{j+1}=\mathcal{Y}_{j+1}^{\perp}$ and $\mathcal{B}_j=\mathcal{Y}_j$. Lemma 3.4 also guarantees that the subspaces $\mathcal{X}_j^{\perp}\cap\mathcal{Y}_j$ and $\mathcal{Y}_{j+1}^{\perp}\cap\mathcal{Y}_j$ contain nonzero elements. Now suppose that $\cos\alpha_j=1$. That is,

$$\sup\left\{\langle u,v\rangle : u\in\mathcal{X}_j^{\perp}\cap\mathcal{Y}_j, v\in\mathcal{Y}_{j+1}^{\perp}\cap\mathcal{Y}_j, \|u\|,\|v\|\le 1\right\}=1.$$

For the purposes of the convergence of the algorithm, we can consider ourselves to be working in the finite-dimensional space spanned by f and the elements of $\{K(\cdot,y):y\in Y_1\}$. Hence, by compactness, we can find elements $u^*\in\mathcal{X}_j^{\perp}\cap\mathcal{Y}_j$ and $v^*\in\mathcal{Y}_{j+1}^{\perp}\cap\mathcal{Y}_j$ with $\|u^*\|=\|v^*\|=1$ such that $\langle u^*,v^*\rangle=1$. By the condition for equality in the Cauchy–Schwarz inequality, it follows that $u^*=v^*$. Thus, $u^*\in\mathcal{X}_j^{\perp}\cap\mathcal{Y}_{j+1}^{\perp}=\mathcal{Y}_j^{\perp}$. However, u^* is also in $\mathcal{Y}_j$. Hence $u^*=0$. This contradicts the fact that $\|u^*\|=1$, and so $\sin\alpha_j\ne 0$ for $j=1,\dots,k$. ∎

We now complement the analysis of Theorem 3.5 by showing how to calculate the angles between subspaces employed in the estimate of the constant c and in the alternative estimates of [5, 7].

Recall from the proof of Theorem 3.5 that α_j is the angle between $\mathcal{X}_j^{\perp}$ and $\mathcal{Y}_{j+1}^{\perp}$. The following result can be found in [3].

LEMMA 3.6. *Let $\mathcal{U}$, $\mathcal{V}$ be closed subspaces of a Hilbert space $\mathcal{H}$. Then the angle between $\mathcal{U}$ and $\mathcal{V}$ is the same as the angle between $\mathcal{U}^{\perp}$ and $\mathcal{V}^{\perp}$.*

This allows us to compute the α_j as the angle between

$$\mathcal{X}_j=\left\{f\in\mathcal{H}: f=\sum_{x\in X_j}\lambda_x K(\cdot,x)\right\}$$

and

$$\mathcal{Y}_{j+1} = \left\{ f \in \mathcal{H} : f = \sum_{y \in Y_{j+1}} \lambda_y K(\cdot, y) \right\}.$$

Relabeling if necessary, we enumerate X_j as $\{x_1, \dots, x_{m_2}\}$ and Y_{j+1} as $\{x_{m_1+1}, \dots, x_{m_3}\}$, where $1 \le m_1 < m_2 \le m_3$. This enumeration is made so that $X_j \cap Y_{j+1} = \{x_{m_1+1}, \dots, x_{m_2}\}$. We now work entirely in the finite-dimensional Hilbert space

$$\mathcal{H} = \left\{ \sum_{j=1}^{m_3} \lambda_j K(\cdot, x_j) : \lambda_j \in \mathcal{R}, j = 1, \dots, m_3 \right\}.$$

We adopt the ";" notation for stacking rows of partitioned vectors. Then any vector $\boldsymbol{\lambda} \in \mathcal{R}^{m_3}$ of coefficients can be partitioned into $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1; \boldsymbol{\lambda}_2; \boldsymbol{\lambda}_3)$, where $\boldsymbol{\lambda}_1 \in \mathcal{R}^{m_1}$, $\boldsymbol{\lambda}_2 \in \mathcal{R}^{m_2-m_1}$, and $\boldsymbol{\lambda}_3 \in \mathcal{R}^{m_3-m_2}$. The corresponding partitioning of the matrix $\mathsf{K} = \big(K(x_i, x_j)\big)_{i,j=1}^{m_3}$ is

$$\begin{array}{cc} \begin{array}{ccc} m_1 & m_2 - m_1 & m_3 - m_2 \end{array} & \\ \begin{pmatrix} \mathsf{D}_1 & \mathsf{E}_1 & \mathsf{F}_1 \\ \mathsf{C}_2 & \mathsf{D}_2 & \mathsf{E}_2 \\ \mathsf{B}_3 & \mathsf{C}_3 & \mathsf{D}_3 \end{pmatrix} & \begin{array}{c} m_1 \\ m_2 - m_1 \\ m_3 - m_2 \end{array} \end{array} .$$

Also the inner product defined in (3.1) of two functions u, v in $\mathcal{H}$ is $\mu^T \mathsf{K} \nu$, where μ and ν are the coefficient vectors of u and v with respect to the reproducing kernel basis.

Now let $u \in \mathcal{X}_j \cap (\mathcal{X}_j \cap \mathcal{Y}_{j+1})^\perp$. Then u has a coefficient vector of the form $(\boldsymbol{\mu}_1; \boldsymbol{\mu}_2; \mathbf{0})$ because $u \in \mathcal{X}_j$. However, since $u \in (\mathcal{X}_j \cap \mathcal{Y}_{j+1})^\perp$ we have $u(x_i) = 0$, $i = m_1 + 1, \dots, m_2$. Hence, u satisfies interpolation equations of the form $\mathsf{K}(\boldsymbol{\mu}_1; \boldsymbol{\mu}_2; \mathbf{0}) = (\boldsymbol{a}; \mathbf{0}; \boldsymbol{b})$, which imply

$$\mathsf{C}_2 \boldsymbol{\mu}_1 + \mathsf{D}_2 \boldsymbol{\mu}_2 = \mathbf{0}.$$

Since D_2 is invertible, $\boldsymbol{\mu}_2 = -\mathsf{D}_2^{-1} \mathsf{C}_2 \boldsymbol{\mu}_1$. Thus any element $u \in \mathcal{X}_j \cap (\mathcal{X}_j \cap \mathcal{Y}_{j+1})^\perp$ has coefficient vector $(\boldsymbol{\mu}_1; \boldsymbol{\mu}_2; \mathbf{0})$, where

$$\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} = \begin{pmatrix} \mathsf{I} \\ -\mathsf{D}_2^{-1} \mathsf{C}_2 \end{pmatrix} \boldsymbol{\mu}_1,$$

$\boldsymbol{\mu}_1 \in \mathcal{R}^{m_1}$, $\boldsymbol{\nu}_2 \in \mathcal{R}^{m_2 - m_1}$, and I is the $m_1 \times m_1$ identity. Note in particular that the matrix on the right of the equation above has full rank.

In our computation of the angle between $\mathcal{X}_j$ and $\mathcal{Y}_{j+1}$, we need to restrict attention to $u \in \mathcal{X}_j$ with $\|u\| = 1$. Now,

$$\begin{aligned} \|u\|^2 &= \begin{pmatrix} \boldsymbol{\mu}_1^T & \boldsymbol{\mu}_2^T & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} \mathsf{D}_1 & \mathsf{E}_1 & \mathsf{F}_1 \\ \mathsf{C}_2 & \mathsf{D}_2 & \mathsf{E}_2 \\ \mathsf{B}_3 & \mathsf{C}_3 & \mathsf{D}_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \mathbf{0} \end{pmatrix} \\ &= \boldsymbol{\mu}_1^T \begin{pmatrix} I & \left(-\mathsf{D}_2^{-1} \mathsf{C}_2\right)^T \end{pmatrix} \begin{pmatrix} \mathsf{D}_1 & \mathsf{E}_1 \\ \mathsf{C}_2 & \mathsf{D}_2 \end{pmatrix} \begin{pmatrix} \mathsf{I} \\ -\mathsf{D}_2^{-1} \mathsf{C}_2 \end{pmatrix} \boldsymbol{\mu}_1 \\ &= \boldsymbol{\mu}_1^T \left(\mathsf{D}_1 - \mathsf{E}_1 \mathsf{D}_2^{-1} \mathsf{C}_2 \right) \boldsymbol{\mu}_1 \\ &= \boldsymbol{\mu}_1^T \left(\mathsf{D}_1 - \mathsf{C}_2^T \mathsf{D}_2^{-1} \mathsf{C}_2 \right) \boldsymbol{\mu}_1 . \end{aligned}$$

The matrix $\left(\begin{smallmatrix}\mathsf{D}_1 & \mathsf{E}_1\\ \mathsf{C}_2 & \mathsf{D}_2\end{smallmatrix}\right)$ is the interpolation matrix for the set X_j and so is strictly positive definite and symmetric. Since $\begin{pmatrix}\mathsf{I} & (-\mathsf{D}_2^{-1}\mathsf{C}_2)^T\end{pmatrix}$ is of full rank we can write $\|u\|^2 = \boldsymbol{\mu}_1^T \mathsf{G}_1 \boldsymbol{\mu}_1$, where G_1 is strictly positive definite and symmetric. Let the Cholesky decomposition of G_1 be $\mathsf{L}_1\mathsf{L}_1^T$. Then $\|u\|^2 = \boldsymbol{\mu}_1^T \mathsf{L}_1\mathsf{L}_1^T \boldsymbol{\mu}_1$. Letting $\boldsymbol{\gamma}_1 = \mathsf{L}_1^T\boldsymbol{\mu}_1$ gives $\|u\|^2 = \boldsymbol{\gamma}_1^T\boldsymbol{\gamma}_1$.

Similar arguments show that if $v \in \mathcal{Y}_{j+1} \cap (\mathcal{X}_j \cap \mathcal{Y}_{j+1})^{\perp}$, then v has coefficient vector $(\mathbf{0}; \boldsymbol{\nu}_2; \boldsymbol{\nu}_3)$. Furthermore, $\|v\|^2$ can be realized via the expression $\|v\|^2 = \boldsymbol{\gamma}_3^T\boldsymbol{\gamma}_3 = \boldsymbol{\nu}_3 \mathsf{L}_3\mathsf{L}_3^T \boldsymbol{\nu}_3$, where $\mathsf{L}_3\mathsf{L}_3^T$ is the Cholesky decomposition of the matrix

$$\begin{pmatrix}\left(-\mathsf{D}_2^{-1}\mathsf{E}_2\right)^T & \mathsf{I}\end{pmatrix}\begin{pmatrix}\mathsf{D}_2 & \mathsf{E}_2\\ \mathsf{C}_3 & \mathsf{D}_3\end{pmatrix}\begin{pmatrix}-\mathsf{D}_2^{-1}\mathsf{E}_2\\ \mathsf{I}\end{pmatrix} = \mathsf{D}_3 - \mathsf{C}_3\mathsf{D}_2^{-1}\mathsf{C}_3^T.$$

Thus the calculation of the cosine of the angle between $\mathcal{X}_j$ and $\mathcal{Y}_{j+1}$ involves finding the supremum over vectors $\boldsymbol{\gamma}_1 \in \mathcal{R}^{m_1}$ and $\boldsymbol{\gamma}_3 \in \mathcal{R}^{m_3-m_2}$ with $\|\boldsymbol{\gamma}_1\|_2 = \|\boldsymbol{\gamma}_3\|_2 = 1$ of the expression

$$\begin{aligned}\mu^T \mathsf{K}\nu &= \boldsymbol{\mu}_1^T \begin{pmatrix}\mathsf{I} & \left(-\mathsf{D}_2^{-1}\mathsf{C}_2\right)^T & \mathsf{O}\end{pmatrix}\begin{pmatrix}\mathsf{D}_1 & \mathsf{E}_1 & \mathsf{F}_1\\ \mathsf{C}_2 & \mathsf{D}_2 & \mathsf{E}_2\\ \mathsf{B}_3 & \mathsf{C}_3 & \mathsf{D}_3\end{pmatrix}\begin{pmatrix}\mathsf{O}\\ -\mathsf{D}_2^{-1}\mathsf{E}_2\\ \mathsf{I}\end{pmatrix}\boldsymbol{\nu}_3\\ &= \boldsymbol{\mu}_1^T \begin{pmatrix}\mathsf{I} & \left(-\mathsf{D}_2^{-1}\mathsf{C}_2\right)^T & \mathsf{O}\end{pmatrix}\begin{pmatrix}-\mathsf{E}_1\mathsf{D}_2^{-1}\mathsf{E}_2 + \mathsf{F}_1\\ \mathsf{O}\\ -\mathsf{C}_3\mathsf{D}_2^{-1}\mathsf{E}_2 + \mathsf{D}_3\end{pmatrix}\boldsymbol{\nu}_3\\ &= \boldsymbol{\mu}_1^T(\mathsf{F}_1 - \mathsf{E}_1\mathsf{D}_2^{-1}\mathsf{E}_2)\boldsymbol{\nu}_3\\ &= \boldsymbol{\gamma}_1^T\mathsf{L}_1^{-1}(\mathsf{F}_1 - \mathsf{E}_1\mathsf{D}_2^{-1}\mathsf{E}_2)\left(\mathsf{L}_3^T\right)^{-1}\boldsymbol{\gamma}_3.\end{aligned}$$

We have therefore established the following result.

THEOREM 3.7. *Adopt the notation prior to the theorem. Then the cosine of the angle between* $\mathcal{X}_j^{\perp}$ *and* $\cap_{i=j+1}^{k}\mathcal{X}_i^{\perp}$ *is given by the* ℓ_2 *norm of the matrix* $\mathsf{L}_1^{-1}(\mathsf{F}_1 - \mathsf{E}_1\mathsf{D}_2^{-1}\mathsf{E}_2)\left(\mathsf{L}_3^T\right)^{-1}$.

An easy adjustment needs to be made to the above theorem if there is no overlap between the sets X_j and Y_{j+1}. In this case, we write the block matrix as

$$\mathsf{K} = \begin{pmatrix}\mathsf{D}_1 & \mathsf{F}_1\\ \mathsf{B}_3 & \mathsf{D}_3\end{pmatrix},$$

and find that the angle between the subspaces $\mathcal{X}_j$ and $\mathcal{Y}_{j+1}$ is the ℓ_2 norm of the matrix $\mathsf{L}_1^{-1}\mathsf{F}_1(\mathsf{L}_3^T)^{-1}$, where $\mathsf{L}_1\mathsf{L}_1^T = \mathsf{D}_1$ and $\mathsf{L}_3\mathsf{L}_3^T = \mathsf{D}_3$.

4. Numerical results of a domain decomposition code. In this section we will briefly discuss some aspects of the implementation and performance of a domain decomposition interpolation code employing some of the previous mathematics. As in section 3, it will be convenient to denote a finite set of distinct points by X and the space of functions "carried" by that set by $\mathcal{X}$.

We list below what we see as the essential ingredients of a domain decomposition code for fitting a globally supported radial basis function by interpolation.

Essential ingredients for a domain decomposition interpolatory fitter.

(i) A method for subdividing space.
(ii) An efficient and scale independent method for solving small interpolation subproblems. The solutions to the small problems will be used to precondition, or approximately solve, the large problem.

(iii) A fast method for computing the action of the large interpolation matrices that occur at various scales.

(iv) An outer iteration.

The method for subdividing space of item (i) above will be used to form the subdomains. Currently we are using a point ordering related to a balanced nD-tree [17] to subdivide space into rectangular boxes. Given that the solution of the large system is to be built upon the solution of many small systems, the need for item (ii) above is clear. This need is filled by, and indeed partly motivated, section 2 of this paper. All matrix iterative methods of which we are aware require at least one matrix-vector product per iteration. Therefore, a prerequisite for the overall method to achieve convergence in $\mathcal{O}(N)$, or $\mathcal{O}(N \log N)$, operations is that a single matrix-vector product costs at most $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$ operations. Thus, for large interpolation problems with globally supported radial basis functions, item (iii) in the list above, a fast way of computing the action of the interpolation matrix on a vector is absolutely essential. Fortunately, such fast evaluators are becoming available for more and more functions Φ. Item (iv) of the list consists of any suitable means of updating the current approximation to the interpolant using the current residuals and quantities derived from the residuals by interpolation on subdomains. It may be as simple as incrementing the current approximation to the interpolant by an approximate interpolant to the current residual.

A simplified two-level code built upon the parts specified in the ingredients list is given below. The method described is a variant of additive Schwarz as opposed to the multiplicative Schwarz method analyzed in section 3.

A simplified domain decomposition interpolation code.

INPUT

Input the finite node set X, the right-hand side f, and the desired accuracy ϵ.

SETUP

(1) Subdivide space (that is, X) into overlapping subdomains $\{X_j : j = 1, \dots, m\}$. For each subdomain classify some points as inner points and some as outer, such that the union of all the inner points is the whole node set.

(2) Choose a coarse grid $Y^{(2)}$ containing some points from each inner subdomain.

(3) Form and factor the matrices required to solve the radial basis function (RBF) interpolation problems on the subdomains. Throughout, r_g, s_g, λ_g and c_g will denote the current residual, the current approximation to the interpolant and parameters of the current approximation to the interpolant at the finest, or global, level.

(5) $r_g \leftarrow f$, $s_g \leftarrow 0$, $\lambda_g \leftarrow 0$, and $c_g \leftarrow 0$.

ITERATIVE SOLUTION

(1) **while** $\|r_g\| > \epsilon$

(2) $\lambda \leftarrow 0$. λ is the coefficient vector of the fine level correction.

(3) **for** $j = 1$ **to** m
(4) Set the coefficients λ_i corresponding to inner points of X_j to the coefficients of the interpolant from the spline space $\mathcal{X}_j$ to $r_g|_{X_j}$.
(5) **end for**
(6) Correct λ to be orthogonal to π_{k-1}.
(7) $s_1 \leftarrow \sum \lambda_j \Phi(\cdot, x_j)$.
(8) Evaluate the residual $r_1 \leftarrow r_g - s_1$ at the coarse grid points.
(9) Interpolate to r_1 at the coarse grid points $Y^{(2)}$ using a spline s_2 (including the polynomial part) from $\mathcal{Y}^{(2)}$.
(10) $s_g \leftarrow s_g + s_1 + s_2$.
(11) Reevaluate the global residuals $r_g \leftarrow f - s_g$ at all the points of X.
(12) **end while**

Clearly there are many choices here. For example, one can use a multiplicative Schwarz (block Gauss–Seidel) like update, rather than the additive Schwarz (block Jacobi) like update of the coefficient vector λ. The analysis of section 3 guarantees that such a multiplicative Schwarz updated version will converge. Note for the analysis to apply one must at step 4 of the iterative solution update all the λ_i's corresponding to $\mathcal{X}_j$, not just those for inner points. As is well known [24], an advantage of additive Schwarz is that it is easier to parallelize as the changes to blocks of coefficient vectors can be made in parallel. Furthermore, in our application, the evaluation of the residuals, typically performed with a variant of the fast multipole method, is also easily parallelized.

Other choices occur in the evaluation of the residuals. There is usually more than one applicable fast evaluation algorithm, and even when the overall fast evaluation algorithm is fixed, there are many detailed implementation decisions to make. Another possibility is to view the forming of the correction function $s_1 + s_2$ as the action of a preconditioning matrix B on the current residual vector r_g. It is then natural to employ some matrix iterative method, for example GMRES, in the outer iteration. Furthermore, for really large data sets one clearly employs more than two levels.

A single pass through the main while loop of the iterative process sketched above can be viewed as a linear process on the input residuals. Therefore, one can represent the process as a mapping $r_g \to \mathsf{R} r_g$. The convergence of the iteration will then be characterized by the spectral radius of this *residual matrix* R. Numerical experiments show that if coarse grid correction is absent, then even with subdomain overlap, the spectral radius of R can be large. Further, these experiments show that the eigenvectors corresponding to the largest magnitude eigenvalue typically represent a low frequency oscillation. Correction by interpolation at coarse grid points can be expected to damp out such low frequency oscillations. Experiments involving calculation of the spectral radius of R, with coarse grid correction, show that this expectation is realized. Specifically calculations of the spectral radius of R have been performed for small model problems and the thin-plate spline in two dimensions. The results are tabulated in Table 3. They show that, as expected, the spectral radius of R is generally small and becomes smaller as the amount of overlap, or the size of the correction grid, is increased.

Table 3

Spectral radius of the residual matrix for four subdomains and various values of overlap and correction size. The grids are all uniform subdivisions of the unit square. The interpolant considered is the thin-plate spline.

Correction grid	Fine grid	Number of overlap rows/columns		
		2	3	4
3×3	10×10	1.391(-1)	9.152(-2)	5.290(-2)
	20×20	1.769(-1)	1.463(-1)	1.355(-1)
	40×40	3.338(-1)	1.823(-1)	1.755(-1)
4×4	10×10	5.606(-2)	4.186(-2)	3.374(-2)
	20×20	2.479(-1)	5.758(-2)	5.145(-2)
	40×40	6.975(-1)	1.556(-1)	6.404(-2)

Table 4

Performance of the current C domain decomposition implementation. The table gives times in seconds to fit the Franke 1 *function until the residual is of magnitude less than* 1.0×10^{-6}. *All computations were carried out using doubles (*64 *bit reals). The node sets were randomly distributed in* $[0,1]^2$ *and the timings performed on a generic machine with an Intel Celeron processor running at* 450 *MHz.*

Number of nodes	Number of level 1 iterations	Time taken in seconds
10,000	8	7.0
20,000	8	17.5
40,000	6	35.5
80,000	6	105.7
160,000	7	407.8

The current C implementation is specialized to fitting 2- and 3-dimensional polyharmonic splines. This was due mostly to the suitability of these splines for some large geophysical and image processing problems motivating our work. A subsidiary reason was that we had in place fast evaluation codes for these functions. However, the ideas and the mathematics of the previous sections apply in the much more general setting of interpolants based upon sums of translates of any strictly conditionally positive definite function Φ. Furthermore, fast evaluation schemes are now being developed for a wide variety of basic functions. Thus we expect the domain decomposition approach to have wide applicability. The existing code has been successfully used to fit data sets of up to 5 million points in 2 dimensions, and up to 250,000 points in 3 dimensions. It is being steadily improved in sophistication, speed, and memory footprint.

The $\mathcal{O}(N^2)$ storage requirements, and $\mathcal{O}(N^3)$ operation counts of direct methods, imply that direct solution of problems with greater than 10,000 centers will be very time consuming, even on very well-endowed machines. Indeed, as late as 1992, many authors have commented on the impracticality of direct, or even iterative, solution of such large RBF interpolation problems. See the quotes in [1]. Bearing this history in mind, the timings in Table 4 are highly satisfactory. They are for interpolation to the

Franke 1 function

$$F^{(1)}(\xi,\eta) = \frac{3}{4}\exp\left(-\frac{(9\xi-2)^2+(9\eta-2)^2}{4}\right) + \frac{3}{4}\exp\left(-\frac{(9\xi+1)^2}{49} - \frac{9\eta+1}{10}\right)$$
$$+\frac{1}{2}\exp\left(-\frac{(9\xi-7)^2+(9\eta-3)^2}{4}\right) - \frac{1}{5}\exp\left(-(9\xi-4)^2-(9\eta-7)^2\right).$$

To the best of our knowledge these timings are superior at the time of writing to those achieved by any competing RBF fitting code. The closest competitors in terms of performance on large problems are the preconditioned GMRES iterative methods discussed in [1]. These are slower but much easier to implement. Numerical experience with the domain decomposition code seems to indicate an approximately $\mathcal{O}(N \log N)$ complexity.

Acknowledgment. We are grateful to Tim Mitchell of the University of Canterbury for his careful efforts in coding various versions of the method.

REFERENCES

[1] R.K. Beatson, J.B. Cherrie, and C.T. Mouat, *Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration*, Adv. Comput. Math., 11 (1999), pp. 253–270.

[2] T.F. Chan and T.P. Mathew, *Domain decomposition algorithms*, in Acta Numer. 1994, Cambridge University Press, 1994, pp. 61–143.

[3] F. Deutsch and H. Hundal, *The rate of convergence of Dykstra's cyclic projections algorithm: The polyhedral case*, Numer. Funct. Anal. Optim., 15 (1994), pp. 537–565.

[4] A.C. Faul and M.J.D. Powell, *Proof of the Convergence of an Iterative Technique for Thin Plate Spline Interpolation in Two Dimensions*, Numerical Analysis Technical Report DAMTP 1998/NA08, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 1998.

[5] C. Hamaker and D.C. Solmon, *The angles between the null spaces of X-rays*, J. Math. Anal. Appl., 62 (1978), pp. 1–23.

[6] S. Kacmarz, *Angenaherte auflosung von systemen linearer gleichungen*, Bull. Acad. Polon. Sci. Lett., A (1937), pp. 355–357.

[7] S. Kayalar and H.L. Weinert, *Error bounds for the method of alternating projections*, Math. Control Signal Systems, 1 (1988), pp. 43–59.

[8] W.A. Light and H.S.J. Wayne, *Spaces of distributions and conditionally positive definite functions*, Numer. Math., 81 (1999), pp. 415–450.

[9] W. Madych and S. Nelson, *Multivariate interpolation and conditionally positive definite functions*, Approx. Theory Appl., 4 (1988), pp. 77–89.

[10] W. Madych, *Miscellaneous error bounds for multiquadric and related interpolators*, Comput. Math. Appl., 24 (1992), pp. 121–138.

[11] C.A. Micchelli, *Interpolation of scattered data: Distance matrices and conditionally positive definite functions*, Constr. Approx., 2 (1986), pp. 11–22.

[12] F. Narcowich and J.D. Ward, *Norms of inverses and condition numbers for matrices associated with scattered data*, J. Approx. Theory, 64 (1991), pp. 69–94.

[13] J. von Neumann, *On rings of operators. Reduction theory*, Ann. of Math. (2), 50 (1949), pp. 401–485.

[14] J. von Neumann, *Functional Operators, Vol.* II*: The Geometry of Orthogonal Spaces*, Princeton University Press, Princeton, NJ, 1950.

[15] G.N. Newsam, *private communication*, Adelaide, Australia, 1996.

[16] M.J.D. Powell, *Some algorithms for thin plate spline interpolation to functions of two variables*, in Adv. Comput. Math., H.P. Dikshit and C.A. Micchelli, eds., World Scientific Publishing Co., New Delhi, India, 1993, pp. 303–319.

[17] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison–Wesley, New York, 1989.

[18] R. Schaback, *Comparison of radial basis function interpolants*, in Multivariate Approximations: From CAGD to Wavelets, K. Jetter and F. Utreras, eds., World Scientific, Singapore, 1993, pp. 293–305.

[19] R. Schaback, *Multivariate interpolation and approximation by translates of a basis function*, in Approximation Theory VIII, Vol. 1, Approximation and Interpolation, C. K. Chui and L. L. Schumacker, eds., World Scientific, Singapore, 1995, pp. 491–514.

[20] R. Schaback and H. Wendland, *Numerical techniques based on radial basis functions*, in Curves and Surface Fitting: Saint-Malo 1999, A. Cohen, C. Rabut, and L.L. Schumaker, eds., Vanderbilt University Press, Nashville, TN, 2000, pp. 359–374.

[21] I.J. Schoenberg, *Metric spaces and completely monotone functions*, Ann. Math. (2), 39 (1938), pp. 811–841.

[22] H.S. Shapiro, *Topics in Approximation Theory*, Lecture Notes in Math., 187, Springer-Verlag, New York, 1971.

[23] R. Sibson and G. Stone, *Computation of thin-plate splines*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1304–1313.

[24] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Partial Differential Equations*, Cambrige University Press, Cambridge, UK, 1996.

[25] K.T. Smith, D.C. Solmon, and S.L. Wagner, *Practical and mathematical aspects of the problem of reconstructing objects from radiographs*, Bull Amer. Math. Soc., 83 (1977), pp. 1227–1270.

[26] H.J. Wenz, *Projection methods for solving interpolation problems with radial basis functions*, presented to Curves and Surfaces, Saint-Malo, France, 1999.