DAMTP 2002/NA05

A Krylov subspace algorithm for multiquadric interpolation in many dimensions

A.C. Faul, G. Goodsell and M.J.D. Powell

Abstract: We consider the version of multiquadric interpolation where the interpolation conditions are the equations $s(\underline{x}_i)=f_i$, $i=1,2,\ldots,n$, and where the interpolant has the form $s(\underline{x})=\sum_{j=1}^{n}\lambda_j\,(\|\underline{x}-\underline{x}_j\|^2+c^2)^{1/2}+\alpha$, $\underline{x}\in\mathcal{R}^d$, subject to the constraint $\sum_{j=1}^{n}\lambda_j=0$. The points $\underline{x}_i\in\mathcal{R}^d$, the right hand sides f_i, $i=1,2,\ldots,n$, and the constant c are data. The equations and the constraint define the parameters λ_j, $j=1,2,\ldots,n$, and α. The resultant approximation $s\approx f$ is useful in many applications, but the calculation of the parameters by direct methods requires $\mathcal{O}(n^3)$ operations, and n may be large. Therefore iterative procedures for this calculation have been studied at Cambridge since 1993, the main task of each iteration being the computation of $s(\underline{x}_i)$, $i=1,2,\ldots,n$, for trial values of the required parameters. These procedures are based on approximations to Lagrange functions, and often they perform very well. For example, ten iterations usually provide enough accuracy in the case $d=2$ and $c=0$, for general positions of the data points, but the efficiency deteriorates if d and c are increased. Convergence can be guaranteed by the inclusion of a Krylov subspace technique that employs the native semi-norm of multiquadric functions. An algorithm of this kind is specified, its convergence is proved, and careful attention is given to the choice of the operator that defines the Krylov subspace, which is analogous to pre-conditioning in the conjugate gradient method. Finally, some numerical results are presented and discussed, for values of d and n from the intervals $[2,40]$ and $[200,10000]$, respectively.

Department of Applied Mathematics and Theoretical Physics,
Centre for Mathematical Sciences,
Wilberforce Road,
Cambridge CB3 0WA,
England.

April, 2002 (revised January, 2004).

1. Introduction

Some radial basis function methods are highly useful for interpolation to values of functions of several variables, because there are no restrictions on the number of variables, d say, or on the positions of the data points, except that the points must all be different. We let the data be the values f_i, $i=1,2,\ldots,n$, of a function $f(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, at the points $\underline{x}_i$, $i=1,2,\ldots,n$, so the interpolant $s(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, should satisfy the equations

$$s(\underline{x}_i) = f_i, \qquad i=1,2,\ldots,n. \tag{1.1}$$

Further, we let s have the multiquadric form

$$s(\underline{x}) = \sum_{j=1}^{n} \lambda_j \left(\|\underline{x}-\underline{x}_j\|^2 + c^2\right)^{1/2} + \alpha, \qquad \underline{x}\in\mathcal{R}^d, \tag{1.2}$$

where the values of the real parameters λ_j, $j=1,2,\ldots,n$, and α are calculated, where the vector norm is Euclidean, and where c is a prescribed real constant. The one degree of freedom in the parameters is taken up in the usual way, by imposing the condition

$$\sum_{j=1}^{n} \lambda_j = 0. \tag{1.3}$$

Expressions (1.1)–(1.3) provide the $(n+1)\times(n+1)$ linear system of equations

$$\left(\begin{array}{cccc|c} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1n} & 1 \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2n} & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ \Phi_{n1} & \Phi_{n2} & \cdots & \Phi_{nn} & 1 \\ \hline 1 & 1 & \cdots & 1 & 0 \end{array}\right) \left(\begin{array}{c} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ \hline \alpha \end{array}\right) = \left(\begin{array}{c} f_1 \\ f_2 \\ \vdots \\ f_n \\ \hline 0 \end{array}\right), \tag{1.4}$$

where $\Phi_{ij}=(\|\underline{x}_i-\underline{x}_j\|^2+c^2)^{1/2}$, $1\leq i,j\leq n$. Micchelli (1986) proves that the matrix of this system is nonsingular. Thus, having chosen the constant c, the function (1.2) is defined uniquely by the conditions (1.1) and (1.3). Many applications of this interpolation procedure are surveyed by Hardy (1990).

Starting with some ideas of Beatson and Powell (1994), an iterative algorithm for solving the system (1.4) was developed at Cambridge, the total amount of work being of magnitude n^2 times the number of iterations. It is the subject of this paper. Major improvements have been made over the early version of the algorithm for thin plate spline interpolation, described by Powell (1997), including the addition of a Krylov subspace technique that ensures convergence (Faul and Powell, 2000). The algorithm is faster than direct methods for large n, provided that the number of iterations is substantially less than n, because, due to the lack of sparsity in the system (1.4), the work of direct methods is $\mathcal{O}(n^3)$. Experiments in practice show favourable iteration counts, especially when d is 2 or 3, or when the data points lie on a smooth 2 or 3 dimensional manifold in many dimensions (Powell, 2002).

The $\mathcal{O}(n^2)$ work of each iteration is due entirely to the calculation of the product $\Phi\underline{\delta}^{(k)}$ for some vector $\underline{\delta}^{(k)}$ in $\mathcal{R}^n$, where Φ is the leading $n\times n$ matrix on the left hand side of equation (1.4). Therefore large increases in n are sometimes made possible by the methods of Cherrie, Beatson and Newsam (2002), which estimate the product in only $\mathcal{O}(n)$ or $\mathcal{O}(n\log n)$ operations. The savings come from the remark that the components of $\Phi\underline{\delta}^{(k)}$ are the values of the function

$$d^{(k)}(\underline{x}) = \sum_{j=1}^{n} \delta_j^{(k)} \left(\|\underline{x}-\underline{x}_j\|^2 + c^2\right)^{1/2}, \qquad \underline{x}\in\mathcal{R}^d, \tag{1.5}$$

at the data points $\underline{x}_i$, $i=1,2,\ldots,n$, where $\delta_j^{(k)}$ is the j-th component of $\underline{\delta}^{(k)}$. Specifically, for each $\underline{x}$, the methods partition the sum (1.5) so that, within each partition, all the distances between the points $\underline{x}_j$ are much less than all the values of $\|\underline{x}-\underline{x}_j\|$. Thus the contribution to $d^{(k)}(\underline{x})$ from a partition is approximated by an expression that can usually be calculated far more quickly than the original sum of the partition. Further, most of these expressions are employed many times as $\underline{x}$ runs through the set $\{\underline{x}_i : i=1,2,\ldots,n\}$. Such techniques allow the system (1.4) to be solved to prescribed accuracy for huge values of n, provided that suitable partitions can be found, but this task becomes difficult or impossible for $d>3$. A fast product of this kind could be added to the iterative algorithm that we study.

Section 2 provides a specification of the algorithm, which is brief, because the reasons for the choices that are made, and many of the details, are given later. The additional material is divided into the following three parts. Section 3 addresses the preliminary operations that are completed before the iterations are begun. Section 4 explains the Krylov subspace features of the algorithm, which are taken from Faul and Powell (2000). They depend on the semi-norm of functions of the form (1.2), subject to the constraint (1.3), that is considered by Schaback (1993). Thus, as explained in Faul (2001), the algorithm can be regarded as a preconditioned conjugate gradient method, the construction of the preconditioner and its properties being the subject of Section 5. Finally, the results of several numerical experiments are presented and discussed in Section 6. They show some advantages and limitations of the algorithm.

2. Specification of the algorithm

The algorithm solves the equations (1.4) to prescribed accuracy. We let $\lambda_j^{(k)}$, $j=1,2,\ldots,n$, and α_k be the estimates of the required parameters λ_j, $j=1,2,\ldots,n$, and α at the beginning of the k-th iteration, the constraint $\sum_{j=1}^{n}\lambda_j^{(k)}=0$ being observed for all k. Further, we let $s^{(k)}$ be the multiquadric function

$$s^{(k)}(\underline{x}) = \sum_{j=1}^{n} \lambda_j^{(k)} \left(\|\underline{x}-\underline{x}_j\|^2 + c^2 \right)^{1/2} + \alpha_k, \qquad \underline{x} \in \mathcal{R}^d, \tag{2.1}$$

so, making the approximation $s^{(k)} \approx s$, the equations (1.1) have the residuals

$$r_i^{(k)} = f_i - s^{(k)}(\underline{x}_i), \qquad i=1,2,\ldots,n. \tag{2.2}$$

The sequence of iterations is terminated when the conditions

$$|r_i^{(k)}| \le \varepsilon, \qquad i=1,2,\ldots,n, \tag{2.3}$$

are achieved for some k, where ε is a given positive tolerance. Another number that has to be prescribed is an integer $q \ge 2$, the value $q=30$ being typical. We find later that its purpose is that some Lagrange functions of the interpolation problem are approximated by functions of the form

$$\widehat{z}_\ell(\underline{x}) = \sum_{j \in \mathcal{L}_\ell} \zeta_{\ell j} \left(\|\underline{x}-\underline{x}_j\|^2 + c^2 \right)^{1/2} + \text{constant}, \qquad \underline{x} \in \mathcal{R}^d, \tag{2.4}$$

where the set $\mathcal{L}_\ell$ contains at most q of the first n integers. The remarks so far introduce most of the notation that occurs in the following description of the algorithm.

Step 0: The values f_i, $i=1,2,\ldots,n$, the data points $\underline{x}_i \in \mathcal{R}^d$, $i=1,2,\ldots,n$, the multiquadric parameter c of expressions (1.2) and (2.1), and the integer q that has just been mentioned are supplied by the user. The choice $c=0$ is allowed.

Step 1: The parameters of the function (2.1) for the first iteration are given the values

$$\left.\begin{array}{l} \lambda_j^{(1)} = 0, \qquad j=1,2,\ldots,n, \qquad \text{and} \\ \alpha_1 = \frac{1}{2}\Big(\min\{f_1, f_2, \ldots, f_n\} + \max\{f_1, f_2, \ldots, f_n\} \Big) \end{array}\right\}, \tag{2.5}$$

and also the initial residuals

$$r_i^{(1)} = f_i - s^{(1)}(\underline{x}_i) = f_i - \alpha_1, \qquad i=1,2,\ldots,n, \tag{2.6}$$

are stored. Termination occurs if condition (2.3) holds for $k=1$, but otherwise the next three steps prepare for the sequence of iterations.

Step 2: A random ordering $\Omega(i)$, $i=1,2,\ldots,n$, of the indices $\{1,2,\ldots,n\}$ is generated. Then the integer m is set to one.

Step 3: The subscript ℓ of equation (2.4) is given the value $\ell=\Omega(m)$. If $n-m+1$ is greater than q, then $\mathcal{L}_\ell$ is composed of q elements, namely the q indices from the set $\{\Omega(m), \Omega(m{+}1), \ldots, \Omega(n)\}$ that minimize the distances $\|\underline{x}_j - \underline{x}_\ell\|$, $j \in \mathcal{L}_\ell$. The choice of ℓ is changed, however, if it is found that this $\mathcal{L}_\ell$ has the property

$$\|\underline{x}_\beta - \underline{x}_\gamma\| \;<\; \tfrac{1}{2} \min \left\{ \|\underline{x}_j - \underline{x}_\ell\| : j \in \mathcal{L}_\ell \backslash \{\ell\} \right\}, \tag{2.7}$$

where β and γ are elements of $\mathcal{L}_\ell$ that minimize $\|\underline{x}_\beta - \underline{x}_\gamma\|$, $\beta \neq \gamma$. In this case, after swapping β with γ if necessary to give $\|\underline{x}_\beta - \underline{x}_\ell\| \leq \|\underline{x}_\gamma - \underline{x}_\ell\|$, the integer $\widehat{m}$ is defined by $\Omega(\widehat{m}) = \beta$, and the values of $\Omega(m)$ and $\Omega(\widehat{m})$ are exchanged. Then the operations of the current step are restarted, the new subscript ℓ being the new value of $\Omega(m)$, which is β. This procedure is finite, because each cycle reduces the right hand side of expression (2.7) by at least a factor of two. Alternatively, when $n{-}m{+}1 \leq q$ occurs, then the set $\mathcal{L}_\ell = \{\Omega(m), \Omega(m{+}1), \ldots, \Omega(n)\}$ is selected.

Step 4: The coefficients $\zeta_{\ell j}$, $j \in \mathcal{L}_\ell$, and the constant of the function (2.4) are calculated by satisfying the constraint $\sum_{j \in \mathcal{L}_\ell} \zeta_{\ell j} = 0$ and the Lagrange conditions

$$\widehat{z}_\ell(\underline{x}_j) = \delta_{j\ell}, \qquad j \in \mathcal{L}_\ell, \tag{2.8}$$

where $\delta_{j\ell}$ is the Kronecker delta. The coefficients are stored, but there is no further need for the constant. If $m \leq n-2$ holds, then m is increased by one, and there is a branch back to Step 3 to find another set $\mathcal{L}_\ell$ and its coefficients. Otherwise, $k=1$ is set, in order to begin the iterations of the algorithm.

Step 5: The coefficients of the function

$$t^{(k)}(\underline{x}) = \sum_{j=1}^{n} \tau_j^{(k)} \left(\|\underline{x} - \underline{x}_j\|^2 + c^2 \right)^{1/2}, \qquad \underline{x} \in \mathcal{R}^d, \tag{2.9}$$

are generated in the following way, beginning with $\tau_j^{(k)} = 0$, $j = 1, 2, \ldots, n$. The index ℓ runs through the set $\{\Omega(1), \Omega(2), \ldots, \Omega(n{-}1)\}$. For each ℓ, the multiplier

$$\mu_\ell^{(k)} = \sum_{i \in \mathcal{L}_\ell} \zeta_{\ell i}\, r_i^{(k)} \Big/ \zeta_{\ell\ell} \tag{2.10}$$

is calculated, the residuals (2.2) being available, and then, for every integer j in $\mathcal{L}_\ell$, the product $\mu_\ell^{(k)} \zeta_{\ell j}$ is added to $\tau_j^{(k)}$. Thus the coefficients of $t^{(k)}$ inherit the property $\sum_{j=1}^{n} \tau_j^{(k)} = 0$ from the constraint $\sum_{j \in \mathcal{L}_\ell} \zeta_{\ell j} = 0$ in Step 4.

Step 6: The coefficients of the search direction

$$d^{(k)}(\underline{x}) = \sum_{j=1}^{n} \delta_j^{(k)} \left(\|\underline{x} - \underline{x}_j\|^2 + c^2 \right)^{1/2}, \qquad \underline{x} \in \mathcal{R}^d, \tag{2.11}$$

are chosen as follows. In the case $k=1$, they are given the values $\delta_j^{(k)} = \tau_j^{(k)}$, $j = 1, 2, \ldots, n$. Otherwise, as in the conjugate gradient method, $d^{(k)}$ is a linear combination of $t^{(k)}$ and $d^{(k-1)}$. Specifically, the formula

$$\delta_j^{(k)} = \tau_j^{(k)} - \beta_k\, \delta_j^{(k-1)}, \qquad j = 1, 2, \ldots, n, \tag{2.12}$$

is applied, where β_k is the ratio

$$\beta_k = \sum_{i=1}^{n} \tau_i^{(k)} d^{(k-1)}(\underline{x}_i) \Big/ \sum_{i=1}^{n} \delta_i^{(k-1)} d^{(k-1)}(\underline{x}_i). \tag{2.13}$$

It follows from the last remark of Step 5 and by induction on k that the coefficients of $d^{(k)}$ also have the property $\sum_{j=1}^{n} \delta_j^{(k)} = 0$.

Step 7: Let $\underline{\delta}^{(k)}$ be the vector in $\mathcal{R}^n$ that has the components $\delta_j^{(k)}$, $j=1,2,\ldots,n$, and let Φ be the leading $n \times n$ matrix on the left hand side of equation (1.4). The product $\Phi\underline{\delta}^{(k)}$ is calculated, because its components are the function values $d^{(k)}(\underline{x}_i)$, $i=1,2,\ldots,n$, as mentioned in the penultimate paragraph of Section 1. These function values are employed in several ways, including their use in formula (2.13) if another iteration is performed.

Step 8: The steplength of the current iteration is set to the ratio

$$\gamma_k = \sum_{i=1}^{n} \delta_i^{(k)} r_i^{(k)} \Big/ \sum_{i=1}^{n} \delta_i^{(k)} d^{(k)}(\underline{x}_i), \tag{2.14}$$

and then we let $s^{(k+1)}$ be the function

$$s^{(k+1)}(\underline{x}) = s^{(k)}(\underline{x}) + \gamma_k d^{(k)}(\underline{x}) + \text{constant}, \qquad \underline{x} \in \mathcal{R}^d, \tag{2.15}$$

where the constant has to be chosen. It follows from equation (2.2) that the new residuals have the form

$$r_i^{(k+1)} = r_i^{(k)} - \gamma_k d^{(k)}(\underline{x}_i) - \text{constant}, \qquad i=1,2,\ldots,n, \tag{2.16}$$

all the numbers on the right hand side being available except the constant. It is given the value that minimizes $\max\{|r_i^{(k+1)}| : i=1,2,\ldots,n\}$, and the residuals (2.16) are stored. Further, α_{k+1} is set to the sum of α_k and the constant, and the multiquadric coefficients of $s^{(k+1)}$ are provided by the formula

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} + \gamma_k \delta_i^{(k)}, \qquad i=1,2,\ldots,n. \tag{2.17}$$

Step 9: The iteration number k is increased by one. Then, if the termination condition (2.3) is not achieved for the new k, there is a branch back to Step 5 for another iteration. Otherwise the calculation is complete, the required parameters being given the values $\lambda_i = \lambda_i^{(k)}$, $i=1,2,\ldots,n$, and $\alpha = \alpha_k$ after the increase in k.

3. The preliminary calculations

Steps 1 to 4 of the algorithm receive further attention in this section. The choice $\lambda_j^{(1)} = 0$, $j=1,2,\ldots,n$, in Step 1 does not prevent one from taking advantage of a good estimate $\widehat{s}$ of the required interpolant. Indeed, if the right hand sides f_i, $i=1,2,\ldots,n$, are replaced by the differences $f_i-\widehat{s}(\underline{x}_i)$, $i=1,2,\ldots,n$, then the algorithm calculates a function s that is a correction to $\widehat{s}$.

The reason for the randomness in Step 2 is to try to avoid orderings of the data points that cause the number of iterations of the algorithm to be unnecessarily large. We address this point, assuming for simplicity that the values $\Omega(i)=i$, $i=1,2,\ldots,n$, occur after Step 3. Hence the set $\mathcal{L}_\ell$, $\ell\in[1,n-1]$, which includes ℓ, is a subset of $\{\ell,\ell+1,\ldots,n\}$. Therefore Step 5 of the algorithm adds $\mu_\ell^{(k)}\zeta_{\ell j}$ to $\tau_j^{(k)}$ only if $j\geq\ell$ holds. Thus the coefficients $\tau_j^{(k)}$, $j=1,2,\ldots,n$, are generated by a procedure that completes the assembly of $\tau_j^{(k)}$ when ℓ reaches the value j. Now it may be advantageous if the construction of $\tau_j^{(k)}$ for small and large indices j is analogous to the use of fine and coarse meshes, respectively, in multigrid methods. The analogy applies if, for typical values of ℓ, the positions of the points $\{\underline{x}_i : i=\ell,\ell+1,\ldots,n\}$ in $\mathcal{R}^d$ are approximately proportional to the original positions $\{\underline{x}_i : i=1,2,\ldots,n\}$. The random ordering of Step 2 is intended to promote this property in bad cases in a way that is easy to implement. For example, a rearrangement of collinear data points is highly useful, if the original ordering is monotonic.

Another advantage of the random ordering is that it assists the identification of the elements of $\mathcal{L}_\ell$ in Step 3. The following method is recommended when $\mathcal{L}_\ell$ is a proper subset of $\{\Omega(m),\Omega(m+1),\ldots,\Omega(n)\}$, the value of $n-m+1$ being greater than q. We let k run through the integers $m+q-1,m+q,\ldots,n$, and, for each k, we let $\mathcal{L}_{\ell k}$ contain the q indices from $\{\Omega(m),\Omega(m+1),\ldots,\Omega(k)\}$ that minimize the distances $\|\underline{x}_j-\underline{x}_\ell\|$, $j\in\mathcal{L}_{\ell k}$. We also let $j(k)$ be an element of $\mathcal{L}_{\ell k}$ that has the property

$$\|\underline{x}_{j(k)}-\underline{x}_\ell\| = \max\{\|\underline{x}_j-\underline{x}_\ell\| : j\in\mathcal{L}_{\ell k}\}. \tag{3.1}$$

It is straightforward to form $\mathcal{L}_{\ell k}$ and $j(k)$ for the initial choice $k=m+q-1$, because in this case $\mathcal{L}_{\ell k}$ is the set $\{\Omega(m),\Omega(m+1),\ldots,\Omega(k)\}$. For each $k<n$, the inequality

$$\|\underline{x}_{\Omega(k+1)}-\underline{x}_\ell\| \geq \|\underline{x}_{j(k)}-\underline{x}_\ell\| \tag{3.2}$$

is tested. If it holds, then $\mathcal{L}_{\ell\, k+1}$ and $j(k+1)$ are just $\mathcal{L}_{\ell k}$ and $j(k)$, respectively, so k is increased by one. Otherwise, $\mathcal{L}_{\ell\, k+1}$ is generated by replacing $j(k)$ by $\Omega(k+1)$ in $\mathcal{L}_{\ell k}$, then $j(k+1)$ is calculated, and k is increased by one as before. By applying this procedure recursively, the algorithm constructs the required set $\mathcal{L}_\ell=\mathcal{L}_{\ell n}$. The amount of work is $\mathcal{O}(q)$ to provide $\mathcal{L}_{\ell k}$ and $j(k)$ initially, it is $\mathcal{O}(1)$ to advance each k that satisfies condition (3.2), and it is $\mathcal{O}(q)$ whenever this condition fails. Further, due to Step 2, the probability of failure of inequality (3.2) is $q/(k-m+2)$. Thus the expected number of operations for the current m is of magnitude

$$(n-m+1) + q\sum_{k=m+q-1}^{n-1}\frac{q}{k-m+2} \approx (n-m+1) + q^2\,[\log(n-m+1)-\log q]. \tag{3.3}$$

Hence the expected work of identifying all the sets $\mathcal{L}_{\Omega(m)}$, $m=1,2,\ldots,n-1$, in Step 3, if no exchanges of $\Omega(m)$ and $\Omega(\widehat{m})$ are made, is bounded by a constant multiple of the expression

$$\begin{aligned}\sum_{m=1}^{n-1}(n-m+1)+q^2\sum_{m=1}^{n-q}\log(n-m+1) \;&\approx\; \tfrac{1}{2}n^2+q^2 n\log n\\ &\le\; \tfrac{1}{2}n^2+2q^2n^{4/3} \;\le\; 2n^2+q^3n,\end{aligned} \tag{3.4}$$

where the last two inequalities are elementary consequences of the positivity of q and n. This amount of work is not excessive, because $\mathcal{O}(n^2)$ and $\mathcal{O}(q^3n)$ operations are required by each iteration and all applications of Step 4, respectively.

The reason for the exchange of $\Omega(m)$ and $\Omega(\widehat{m})$ in Step 3, when $\mathcal{L}_\ell$ has the property (2.7), is that, if $\min\{\|\underline{x}_j-\underline{x}_\ell\| \,:\, j\in\mathcal{L}_\ell\backslash\{\ell\}\}$ is relatively small, then the function (2.4) seems to be good for preconditioning in the conjugate gradient interpretation of the algorithm. An explanation of this remark is given in Section 5. Moreover, the multigrid suggestion in the second paragraph of this section is strengthened if nearest neighbour distances between the points $\underline{x}_{\Omega(j)}$, $j\in[m,n]$, grow as rapidly as possible as m increases. This objective is also assisted by the exchange of values of $\Omega(m)$ and $\Omega(\widehat{m})$. Indeed, the exchange does not alter these distances, but it does provide a substantial reduction in the nearest neighbour distance of the point $\underline{x}_{\Omega(m)}$. Thus, as m increases, points that are relatively close to their neighbours tend to be removed from the set $\{\underline{x}_{\Omega(j)} : j\in[m,n]\}$.

The following direct method is used to solve the multiquadric interpolation equations in Step 4 of Section 2. We describe it using the notation (1.4), assuming for the moment that $\mathcal{L}_\ell$ is the set $\{1,2,\ldots,n\}$, that n is at most q temporarily, and that the components f_i, $i=1,2,\ldots,n$, are the right hand sides of the Lagrange conditions (2.8). It follows that the unknowns λ_j, $j=1,2,\ldots,n$, and α of the system (1.4) are the required multiquadric parameters $\zeta_{\ell j}$, $j\in\mathcal{L}_\ell$, and the constant term of the function (2.4). We form ν new equations, where $\nu=n-1$, by subtracting the n-th equation of the system from the earlier ones, which removes the dependence on α. Further, the last equation is eliminated by defining $\lambda_n=-\sum_{j=1}^{\nu}\lambda_j$. These operations provide the conditions

$$\begin{pmatrix}\Phi_{11}-\Phi_{n1} & \Phi_{12}-\Phi_{n2} & \cdots & \Phi_{1n}-\Phi_{nn}\\ \Phi_{21}-\Phi_{n1} & \Phi_{22}-\Phi_{n2} & \cdots & \Phi_{2n}-\Phi_{nn}\\ \vdots & \vdots & & \vdots\\ \Phi_{\nu 1}-\Phi_{n1} & \Phi_{\nu 2}-\Phi_{n2} & \cdots & \Phi_{\nu n}-\Phi_{nn}\end{pmatrix}\begin{pmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_\nu\\ -\sum_{j=1}^{\nu}\lambda_j\end{pmatrix}=\begin{pmatrix}f_1-f_n\\ f_2-f_n\\ \vdots\\ f_\nu-f_n\end{pmatrix}. \tag{3.5}$$

They are equivalent to the $\nu\times\nu$ system

$$\Psi\,\underline{\lambda} \;=\; \underline{g}, \tag{3.6}$$

where the matrix Ψ has the elements

$$\Psi_{ij} = (\Phi_{ij} - \Phi_{nj}) - (\Phi_{in} - \Phi_{nn}), \qquad 1 \leq i, j \leq \nu, \tag{3.7}$$

where $\underline{\lambda}$ has the components λ_j, $j=1,2,\ldots,\nu$, and where $\underline{g} \in \mathcal{R}^\nu$ is the right hand side of expression (3.5). Thus Ψ is the product $Q^T \Phi\, Q$, if we let Q^T be the $\nu \times n$ matrix

$$Q^T = \left(\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & -1 \\ 0 & 1 & \ddots & \vdots & -1 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -1 \end{array} \right), \tag{3.8}$$

so Ψ inherits symmetry from Φ. Now an important intermediate result in the analysis of the system (1.4) by Micchelli (1986) is that, if $\underline{w}$ is any nonzero vector in $\mathcal{R}^n$ whose components sum to zero, then $\underline{w}^T \Phi\, \underline{w}$ is negative. Further, the vector $\underline{w} = Q\, \underline{v}$ has these properties if $\underline{v}$ is any nonzero vector in $\mathcal{R}^\nu$, which provides the strict inequality

$$0 > \underline{w}^T \Phi\, \underline{w} = \underline{v}^T Q^T \Phi\, Q\, \underline{v} = \underline{v}^T \Psi\, \underline{v}, \qquad \underline{v} \neq 0. \tag{3.9}$$

Therefore Ψ is not only symmetric but also negative definite. The recommended direct method for solving the equations (1.4) calculates the Cholesky factorization of $-\Psi$, and then $\underline{\lambda} \in \mathcal{R}^\nu$ is derived from the system (3.6) by forward and backward substitution. Further, λ_n is given the value $-\sum_{j=1}^{\nu} \lambda_j$, and α is not required. Actually, there is no need for the forward substitution if only the last component of $\underline{g}$ is nonzero, which can be achieved by ordering the elements of $\mathcal{L}_\ell$ so that f_ν in expression (3.5) is the nonzero right hand side of the Lagrange conditions (2.8). Another useful observation is that, in all but one of the numerical experiments so far, nonpositive Cholesky pivots have occurred only when there are coincidences among the data points $\underline{x}_i$, $i=1,2,\ldots,n$, the exception being identified in Section 6. It follows from the remarks of this paragraph that the amount of work of Step 4 for each ℓ is of magnitude $|\mathcal{L}_\ell|^3$, which accounts for the $\mathcal{O}(q^3 n)$ operations that are mentioned soon after inequality (3.4). This operation count should be taken into consideration when the value of q is chosen for Step 0 of the algorithm.

4. Krylov subspace properties

Let $\mathcal{A}$ be the linear space of functions of the form (1.2), where $\underline{x}_i \in \mathcal{R}^d$, $i = 1,2,\ldots,n$, and c are given, and where the parameters λ_j, $j=1,2,\ldots,n$, and α can take any real values subject to the constraint (1.3). Then, because the data points must all be different, the dimension of $\mathcal{A}$ is n. Further, because the matrix of the system (1.4) is nonsingular, the interpolation conditions (1.1) define $s \in \mathcal{A}$

uniquely. We let s^* be the required interpolant, in order to reserve the notation s for a general element of $\mathcal{A}$.

The iterative algorithm of Section 2 is a Krylov subspace method, because, as shown in this section, it has the following properties. Step 6 generates search directions $d^{(k)} \in \mathcal{A}$, $k=1,2,3,\ldots$, that are linearly independent until termination occurs. Therefore, because each direction has the form (2.11) with coefficients that satisfy $\sum_{j=1}^{n} \delta_j^{(k)} = 0$, termination is achieved after at most $n-1$ iterations. Furthermore, there exists a scalar product $\langle \cdot, \cdot \rangle_\phi$ between elements of $\mathcal{A}$, such that the sequence of search directions enjoys the conditions

$$\langle d^{(k)}, d^{(\ell)} \rangle_\phi = 0, \qquad k \neq \ell, \tag{4.1}$$

and also the choice (2.14) of the steplength provides the equations

$$\langle s^* - s^{(k+1)}, d^{(j)} \rangle_\phi = 0, \qquad j=1,2,\ldots,k, \tag{4.2}$$

where $s^{(k+1)}$ is the approximation (2.15) to s^*. These properties are typical of preconditioned conjugate gradient procedures. The Krylov subspace feature is that we choose a linear operator Ξ from $\mathcal{A}$ to $\mathcal{A}$. Then, for each iteration number k, the direction $d^{(k)}$ is in the linear subspace of $\mathcal{A}$ that is spanned by the functions $\Xi^\ell s^*$, $\ell=1,2,\ldots,k$.

The scalar product $\langle \cdot, \cdot \rangle_\phi$ between elements of $\mathcal{A}$ depends on the remark, made soon after equation (3.8), that, if $\underline{w}$ is any nonzero vector in $\mathcal{R}^n$ whose components sum to zero, then $\underline{w}^T \Phi\, \underline{w}$ is negative. It follows from the constraint (1.3) that the equation

$$\|s\|_\phi \;=\; (-\underline{\lambda}^T \Phi \underline{\lambda})^{1/2}, \qquad s \in \mathcal{A}, \tag{4.3}$$

defines a semi-norm of the linear space $\mathcal{A}$, where the components of $\underline{\lambda}$ are the radial basis function coefficients of s as in expression (1.2). We pick the scalar product that is induced by this semi-norm, the value of $\langle s,t \rangle_\phi$ being defined by the identity

$$\|s+t\|_\phi^2 \;=\; \|s\|_\phi^2 + 2\,\langle s,t \rangle_\phi + \|t\|_\phi^2, \qquad s,t \in \mathcal{A}. \tag{4.4}$$

Hence we will deduce later from condition (4.2) that $\|s^* - s^{(k)}\|_\phi$ decreases monotonically as k increases. Expression (4.3) is a semi-norm, because $s \in \mathcal{A}$ and $\|s\|_\phi = 0$ imply that s is a constant function, which need not be zero. Thus the addition of a constant to the right hand sides f_i, $i=1,2,\ldots,n$, makes no difference to the search directions $d^{(k)}$, $k \geq 1$, nor to the value of $\|s^{(k+1)} - s^*\|_\phi$. Therefore the constant part of $s^{(k+1)}$ requires special attention, the method of Step 8 being suitable. This method causes all the residuals (2.16) to be zero if $s^{(k+1)} - s^*$ is a constant function. It follows that Step 5 is reached from Step 9 only if the new value of $\|s^{(k)} - s^*\|_\phi$ is positive.

A strong advantage of the semi-norm (4.3) is that Φ can be eliminated from the formula for scalar products. Specifically, if $s \in \mathcal{A}$ is the function (1.2), and if

$t\in\mathcal{A}$ has the multiquadric coefficients μ_j, $j=1,2,\ldots,n$, which sum to zero, then, using the constraint (1.3), we find that the scalar product of the identity (4.4) has the value

$$\begin{aligned}\langle s,t\rangle_\phi \;&=\; -\sum_{i=1}^n\sum_{j=1}^n \lambda_i\Phi_{ij}\mu_j \;=\; -\sum_{i=1}^n \lambda_i \sum_{j=1}^n \mu_j \left(\|\underline{x}_i-\underline{x}_j\|^2+c^2\right)^{1/2} \\ &=\; -\sum_{i=1}^n \lambda_i\, t(\underline{x}_i) \;=\; -\sum_{j=1}^n \mu_j\, s(\underline{x}_j), \qquad (4.5)\end{aligned}$$

the last equation being due to symmetry. In particular, we deduce from expressions (2.9) and (2.11) that the numerator and denominator of the ratio (2.13) are $-\langle t^{(k)},d^{(k-1)}\rangle_\phi$ and $-\langle d^{(k-1)},d^{(k-1)}\rangle_\phi$, respectively. Thus, because the constant terms of $d^{(k)}$, $t^{(k)}$ and $d^{(k-1)}$ are all zero, formula (2.12) is equivalent to the equation

$$d^{(k)}(\underline{x}) \;=\; t^{(k)}(\underline{x}) - \frac{\langle t^{(k)},d^{(k-1)}\rangle_\phi}{\langle d^{(k-1)},d^{(k-1)}\rangle_\phi}\, d^{(k-1)}(\underline{x}), \qquad \underline{x}\in\mathcal{R}^d. \qquad (4.6)$$

Therefore the choice of β_k provides the conjugacy condition $\langle d^{(k)},d^{(k-1)}\rangle_\phi=0$, $k\geq 2$. Similarly, because $s^*-s^{(k)}$ is an element of $\mathcal{A}$, and because the residuals (2.2) are the values of $s^*-s^{(k)}$ at the interpolation points $\underline{x}_i$, $i=1,2,\ldots,n$, the numerator and denominator of the ratio (2.14) are $-\langle d^{(k)},s^*-s^{(k)}\rangle_\phi$ and $-\langle d^{(k)},d^{(k)}\rangle_\phi$, respectively. Hence formula (2.15) is equivalent to the equation

$$(s^*-s^{(k+1)})(\underline{x}) = (s^*-s^{(k)})(\underline{x}) - \frac{\langle s^*-s^{(k)},d^{(k)}\rangle_\phi}{\langle d^{(k)},d^{(k)}\rangle_\phi}\, d^{(k)}(\underline{x}) - \text{constant}, \quad \underline{x}\in\mathcal{R}^d, \; (4.7)$$

which gives the orthogonality condition $\langle s^*-s^{(k+1)},d^{(k)}\rangle_\phi=0$. In other words, the algorithm picks the steplength $\gamma_k\in\mathcal{R}$ that has the property

$$\|s^*-s^{(k+1)}\|_\phi = \|s^*-s^{(k)}-\gamma_k\, d^{(k)}\|_\phi = \min\{\|s^*-s^{(k)}-\gamma\, d^{(k)}\|_\phi : \gamma\in\mathcal{R}\}. \quad (4.8)$$

Therefore the semi-norms $\|s^*-s^{(k)}\|_\phi$, $k=1,2,3,\ldots$, decrease monotonically as stated after equation (4.4), and we can take the view that γ_k is defined by an optimal line search. We prove later in this section that every search direction satisfies $\|d^{(k)}\|_\phi>0$, so no divisions by zero occur in expressions (4.6) and (4.7).

The operator Ξ from $\mathcal{A}$ to $\mathcal{A}$, mentioned soon after expression (4.2), is defined by the equation

$$(\Xi s)(\underline{x}) \;=\; \sum_{\ell\in\Omega_{\rm all}} \frac{\langle \check{z}_\ell, s\rangle_\phi}{\langle \check{z}_\ell,\check{z}_\ell\rangle_\phi}\, \check{z}_\ell(\underline{x}), \qquad \underline{x}\in\mathcal{R}^d, \quad s\in\mathcal{A}, \qquad (4.9)$$

where $\Omega_{\rm all}$ is the set $\{\Omega(i) : i=1,2,\ldots,n-1\}$, and where $\check{z}_\ell$ is the function (2.4) without its constant term. The construction of the coefficients $\tau_j^{(k)}$, $j=$

$1,2,\ldots,n$, in Step 5 of the algorithm, is designed so that $t^{(k)}$ becomes the function $\Xi\,(s^*-s^{(k)})$, which is proved as follows. Again we employ the remark that $r_i^{(k)}$ is the value of $s^*-s^{(k)}$ at $\underline{x}_i$. Thus the numerator of the ratio (2.10) is the scalar product $-\langle \check{z}_\ell, s^*-s^{(k)}\rangle_\phi$. Further, the Lagrange conditions (2.8) and the constraint $\sum_{j\in\mathcal{L}_\ell}\zeta_{\ell j}=0$ imply that the denominator of the ratio has the value

$$\zeta_{\ell\ell} = \sum_{j\in\mathcal{L}_\ell} \zeta_{\ell j}\,\widehat{z}_\ell(\underline{x}_j) = \sum_{j\in\mathcal{L}_\ell} \zeta_{\ell j}\,\check{z}_\ell(\underline{x}_j) = -\langle \check{z}_\ell, \check{z}_\ell\rangle_\phi, \tag{4.10}$$

which is nonzero, because the Lagrange conditions do not allow $\check{z}_\ell \in \mathcal{A}$ to be constant. Moreover, the addition of $\mu_\ell^{(k)}\zeta_{\ell j}$ to $\tau_j^{(k)}$ for every j in $\mathcal{L}_\ell$, which occurs in Step 5, makes the contribution $\mu_\ell^{(k)}\check{z}_\ell(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, to the function (2.9). It follows that the sum of these contributions is the required function

$$t^{(k)} \;=\; \sum_{i=1}^{n-1} \mu_{\Omega(i)}^{(k)}\,\check{z}_{\Omega(i)} \;=\; \sum_{i=1}^{n-1} \frac{\langle \check{z}_{\Omega(i)}, s^*-s^{(k)}\rangle_\phi}{\langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(i)}\rangle_\phi}\,\check{z}_{\Omega(i)} \;=\; \Xi\,(s^*-s^{(k)}). \tag{4.11}$$

The operator (4.9) has the property that $\Xi\, s$, $s\in\mathcal{A}$, is the zero function if and only if s is constant. The part of this statement when s is a constant is true, because then all of the scalar products $\langle \check{z}_\ell, s\rangle_\phi$ in the sum (4.9) are zero. A proof of the other part of the statement depends on the assertion that any $s\in\mathcal{A}$ can be expressed in the form

$$s(\underline{x}) \;=\; \sum_{\ell\in\Omega_{\rm all}} \theta_\ell\,\check{z}_\ell(\underline{x}) + \text{constant}, \qquad \underline{x}\in\mathcal{R}^d, \tag{4.12}$$

where the following construction gives the multipliers θ_ℓ. We let m run through the integers $1,2,\ldots,n-1$, for each m we set $\ell=\Omega(m)$, and we let $\theta_\ell=\theta_{\Omega(m)}$ have the value that causes the coefficients of the term $(\|\underline{x}-\underline{x}_\ell\|^2+c^2)^{1/2}$, $\underline{x}\in\mathcal{R}^d$, in the functions s and $\sum_{i=1}^m \theta_{\Omega(i)}\check{z}_{\Omega(i)}$ to be the same. Each θ_ℓ is well-defined, because the number (4.10) is nonzero. Further, this equality of coefficients is not disturbed by the construction of the later terms $\sum_{i=m+1}^{n-1}\theta_{\Omega(i)}\check{z}_{\Omega(i)}$, because, according to Step 3 of the algorithm, the set $\mathcal{L}_\ell=\mathcal{L}_{\Omega(m)}$ of expression (2.4) is a subset of the integers $\Omega(i)$, $i=m,m{+}1,\ldots,n$, for each m. Thus equal multiples of $(\|\underline{x}-\underline{x}_\ell\|^2+c^2)^{1/2}$, $\underline{x}\in\mathcal{R}^d$, occur on the two sides of equation (4.12) for $\ell=\Omega(m)$, $m=1,2,\ldots,n-1$. The multipliers are also the same when $\ell=\Omega(n)$, because the multiquadric coefficients of every function under consideration sum to zero. Therefore the expansion (4.12) is valid for some choice of the constant. Further, if $s\in\mathcal{A}$ is not a constant, the strict inequality

$$\|s\|_\phi^2 \;=\; \langle s,s\rangle_\phi \;=\; \sum_{\ell\in\Omega_{\rm all}} \theta_\ell\,\langle \check{z}_\ell, s\rangle_\phi \;>\; 0 \tag{4.13}$$

holds, so at least one of the scalar products $\langle \check{z}_\ell, s\rangle_\phi$, $\ell\in\Omega_{\rm all}$, is nonzero. Thus one or more nonzero terms occur in the sum (4.9). Moreover, we infer from the existence of the expansion (4.12) for general $s \in \mathcal{A}$, and from $\dim \mathcal{A} = n$, that the functions $\check{z}_\ell$, $\ell \in \Omega_{\rm all}$, and a nonzero constant are linearly independent. These remarks imply that, if s satisfies inequality (4.13), then $(\Xi\, s)(\underline{x})$, $\underline{x} \in \mathcal{R}^d$, is not constant. In other words, the choice of Ξ provides the condition

$$\|s\|_\phi > 0 \quad \Longleftrightarrow \quad \|\Xi\, s\|_\phi > 0, \qquad s \in \mathcal{A}. \tag{4.14}$$

Another important feature of the definition (4.9) is the identity

$$\langle t, \Xi s\rangle_\phi \;=\; \sum_{\ell\in\Omega_{\rm all}} \frac{\langle \check{z}_\ell, s\rangle_\phi}{\langle \check{z}_\ell, \check{z}_\ell\rangle_\phi}\, \langle \check{z}_\ell, t\rangle_\phi \;=\; \langle \Xi t, s\rangle_\phi, \qquad s,t \in \mathcal{A}. \tag{4.15}$$

We are now ready for the following analysis, which establishes the properties of the algorithm that are stated in the second paragraph of this section.

Theorem 1: Termination occurs in Step 1 of the algorithm if $\Xi\, s^*$ is the zero function. Otherwise, let κ be the least integer such that the functions $\Xi^\ell s^*$, $\ell = 1, 2, \ldots, \kappa+1$, are linearly dependent. Then termination occurs no later than the κ-th iteration. Further, for each iteration number k, the conditions (4.1) and (4.2) are achieved, the steplength γ_k is positive, and the search directions have the property

$$\text{span}\,\{d^{(j)} : j = 1, 2, \ldots, k\} \;=\; \text{span}\,\{\Xi^\ell s^* : \ell = 1, 2, \ldots, k\}. \tag{4.16}$$

Proof: We have noted already that $\Xi\, s^*$ is zero if and only if $s^* \in \mathcal{A}$ is a constant function, and then the initial residuals (2.6) would be zero. Therefore the first statement of the theorem is true. It remains to consider the case $\kappa > 0$.

We employ induction on k, the hypotheses of the induction being equivalent to conditions (4.1), (4.2) and (4.16). When $k = 1$, the choice of $d^{(1)}$ in Step 6, formula (4.11), and $s^{(1)} =$ constant, give the identities

$$d^{(1)} \;=\; t^{(1)} \;=\; \Xi\,(s^* - s^{(1)}) \;=\; \Xi\, s^*. \tag{4.17}$$

Therefore equation (4.16) holds when $k = 1$, and in the inductive argument we assume it is true for the current and all previous values of k. We write the conditions (4.1) as the conjugacy properties

$$\langle d^{(k-1)}, d^{(\ell)}\rangle_\phi \;=\; 0, \qquad \ell = 1, 2, \ldots, k-2, \tag{4.18}$$

which are vacuous for $k \le 2$, but we assume in the proof that they hold at the beginning of every iteration with $k \ge 3$. Furthermore, we write the conditions (4.2) as the orthogonality properties

$$\langle s^* - s^{(k)}, d^{(j)}\rangle_\phi \;=\; 0, \qquad j = 1, 2, \ldots, k-1, \tag{4.19}$$

which are assumed to hold at the beginning of every iteration with $k \ge 2$. Hence, in order to complete the proof, we show that equations (4.18) and (4.19) remain true if k is replaced by $k{+}1$, that γ_k is positive, and that either termination occurs on the current iteration or the identity (4.16) is inherited by the next value of k, except that this alternative to termination does not happen in the case $k{=}\kappa$.

When $k \le \kappa$, the linear space (4.16) has dimension k. Further, the semi-norm of every nonzero element of this space is positive, because the definition (4.9) shows that the functions $\Xi\, s$, $s \in \mathcal{A}$, are without constant terms. It follows that $\|d^{(k)}\|_\phi$ is positive. Therefore the functions (4.6) and (4.7) are well-defined for $k \ge 2$ and $k \ge 1$, respectively. Moreover, if $k \ge 3$, our inductive treatment of the conjugacy properties (4.18) requires the remark that, by applying Ξ to both sides of equation (4.16) after reducing k by 2, we find the relation

$$\begin{aligned} \mathrm{span}\,\{\Xi\, d^{(j)} : j{=}1,2,\ldots,k{-}2\} \;&=\; \mathrm{span}\,\{\Xi^{\ell+1} s^* : \ell{=}1,2,\ldots,k{-}2\} \\ \subset \mathrm{span}\,\{\Xi^{\ell} s^* : \ell{=}1,2,\ldots,k{-}1\} \;&=\; \mathrm{span}\,\{d^{(j)} : j{=}1,2,\ldots,k{-}1\}. \end{aligned} \tag{4.20}$$

Thus the hypothesis (4.19) provides the values

$$\langle s^*{-}s^{(k)}, \Xi\, d^{(j)} \rangle_\phi \;=\; 0, \qquad j{=}1,2,\ldots,k{-}2. \tag{4.21}$$

Hence equations (4.6), (4.11), (4.18) and (4.15) give the conjugacy conditions

$$\begin{aligned} \langle d^{(k)}, d^{(\ell)} \rangle_\phi &= \langle t^{(k)} - \beta_k\, d^{(k-1)}, d^{(\ell)} \rangle_\phi = \langle \Xi\,(s^*{-}s^{(k)}), d^{(\ell)} \rangle_\phi \\ &= \langle s^*{-}s^{(k)}, \Xi\, d^{(\ell)} \rangle_\phi = 0, \qquad \ell{=}1,2,\ldots,k{-}2, \end{aligned} \tag{4.22}$$

while formula (4.6) implies $\langle d^{(k)}, d^{(k-1)} \rangle_\phi {=} 0$. Therefore the inductive hypothesis (4.18) is preserved if k is replaced by $k{+}1$. This remark with equations (2.15) and (4.19) yield the orthogonality properties

$$\langle s^*{-}s^{(k+1)}, d^{(j)} \rangle_\phi = \langle s^*{-}s^{(k)} - \gamma_k\, d^{(k)}, d^{(j)} \rangle_\phi = 0, \quad j{=}1,2,\ldots,k{-}1, \tag{4.23}$$

while formula (4.7) implies $\langle s^*{-}s^{(k+1)}, d^{(k)} \rangle_\phi {=} 0$. Therefore the inductive hypothesis (4.19) is also preserved if k is replaced by $k{+}1$.

Another consequence of $\langle s^* - s^{(k+1)}, d^{(k)} \rangle_\phi = 0$, with expression (2.15), is that γ_k satisfies the equation

$$\langle s^*, d^{(k)} \rangle_\phi \;=\; \langle s^{(k+1)}, d^{(k)} \rangle_\phi \;=\; \langle s^{(k)}, d^{(k)} \rangle_\phi + \gamma_k\, \langle d^{(k)}, d^{(k)} \rangle_\phi. \tag{4.24}$$

Hence γ_k is positive as required if and only if the scalar product $\langle s^* - s^{(k)}, d^{(k)} \rangle_\phi$ is positive. Formula (4.6), the last of the conditions (4.19), and equations (4.11) and (4.9), give the value

$$\langle s^*{-}s^{(k)}, d^{(k)} \rangle_\phi \;=\; \langle s^*{-}s^{(k)}, t^{(k)} \rangle_\phi \;=\; \sum_{\ell \in \Omega_{\mathrm{all}}} \frac{\langle \check{z}_\ell, s^* - s^{(k)} \rangle_\phi^2}{\langle \check{z}_\ell, \check{z}_\ell \rangle_\phi}, \tag{4.25}$$

which vanishes only if all the scalar products $\langle \check{z}_\ell, s^* - s^{(k)} \rangle_\phi$, $\ell \in \Omega_{\mathrm{all}}$, are zero. In that case, however, the definition (4.9) would imply $\Xi(s^* - s^{(k)}) {=} 0$, and then the property (4.14) would show that $s^* - s^{(k)}$ is a constant function, so the calculation would have been terminated before the beginning of the k-th iteration. Therefore γ_k is positive.

Next we address the case $k=\kappa$. Then the functions $\Xi^\ell s^*$, $\ell=1,2,\ldots,k+1$, are linearly dependent, although the space (4.16) has dimension k. Thus the equation

$$\Xi^{k+1} s^* \;=\; \sum_{\ell=1}^{k} \xi_\ell \, \Xi^\ell s^* \tag{4.26}$$

is satisfied for some multipliers ξ_ℓ, $\ell=1,2,\ldots,k$. They are not all zero, because the positivity of $\|\Xi\, s^*\|_\phi$ with an inductive application of the property (4.14) gives $\|\Xi^{k+1} s^*\|_\phi > 0$. Further, ξ_1 is nonzero, because otherwise the function $\Xi^{k-1} s^* - \sum_{\ell=0}^{k-2} \xi_{\ell+2} \, \Xi^\ell s^*$ would be in the null space of Ξ^2, which is the same as the null space of Ξ, so the dimension of the space (4.16) would be less than k. Therefore equation (4.26) implies that $\Xi\, s^*$ is in the span of the functions $\Xi^\ell s^*$, $\ell=2,3,\ldots,k+1$. It follows that s^* is in the $(k+1)$ dimensional linear space, $\mathcal{S}_k$ say, that includes the space (4.16) and constants. Moreover, because Step 1 of the algorithm sets $s^{(1)}$ to a constant function, the construction (2.15) implies the equation

$$s^{(k+1)}(\underline{x}) \;=\; \sum_{j=1}^{k} \gamma_j \, d^{(j)}(\underline{x}) + \text{constant}, \qquad \underline{x} \in \mathcal{R}^d, \tag{4.27}$$

so $s^{(k+1)}$ is also in $\mathcal{S}_k$. Hence $s^* - s^{(k+1)}$ is an element of $\mathcal{S}_k$, but equation (4.19) with k increased by one shows that $s^* - s^{(k+1)}$ is orthogonal to every element of $\mathcal{S}_k$. Therefore $s^* - s^{(k+1)}$ is also a constant function. It follows that the termination condition of the algorithm is achieved no later than the κ-th iteration.

It remains to establish that the property (4.16) holds when k is increased by one, if termination does not occur on the k-th iteration, in which case the functions $\Xi^\ell s^*$, $\ell=1,2,\ldots,k+1$, are linearly independent. Equation (4.16) allows $d^{(k)}$ to be expressed as the sum

$$d^{(k)}(\underline{x}) \;=\; \sum_{\ell=1}^{k} \omega_\ell \, (\Xi^\ell s^*)(\underline{x}), \qquad \underline{x} \in \mathcal{R}^d, \tag{4.28}$$

where the last of the multipliers ω_ℓ, $\ell=1,2,\ldots,k$, is nonzero, because otherwise the dimension of the space on the left hand side of condition (4.16) would be less than k. Further, expressions (4.27) and (4.28) imply that the function (4.11) of the next iteration can be written in the form

$$t^{(k+1)} \;=\; \Xi\,(s^* - s^{(k+1)}) \;=\; \Xi\, s^* - \sum_{j=1}^{k-1} \gamma_j \, \Xi\, d^{(j)} - \gamma_k \sum_{\ell=1}^{k} \omega_\ell \, \Xi^{\ell+1} s^*. \tag{4.29}$$

We see that all the terms on the right hand side are in the space (4.16), except for $-\gamma_k \, \omega_k \, \Xi^{k+1} s^*$. Moreover, equation (4.6) shows that the difference between $d^{(k+1)}$

and $t^{(k+1)}$ is in the space (4.16). Therefore $d^{(k+1)}$ has the form

$$d^{(k+1)} \;=\; -\gamma_k\,\omega_k\,\Xi^{k+1}s^* + \sum_{\ell=1}^{k} \check{\omega}_\ell\,\Xi^\ell s^*, \tag{4.30}$$

for some multipliers $\check{\omega}_\ell$, $\ell=1,2,\ldots,k$. Now the conditions $\gamma_k>0$ and $\omega_k\neq 0$ have been established already. It follows from equation (4.30) that the identity (4.16) is preserved when k is increased by one, which completes the proof. □

5. On the choice of Ξ

Theorem 1 shows that the algorithm of Section 2 can be regarded as a preconditioned conjugate gradient method for minimizing $\|s-s^*\|_\phi^2$, $s\in\mathcal{A}$, the preconditioning being given by the choice of the operator (4.9). The theorem shows also that termination would occur at the end of the first iteration if $\Xi^2 s^*$ were a multiple of Ξs^*. This condition holds if Ξ has the property $\|\Xi s-s\|_\phi=0$, $s\in\mathcal{A}$, because then Ξs^*-s^* would be constant, which would imply $\Xi^2 s^*=\Xi s^*$. Usually, however, more than one iteration is required. On the other hand, we take the view that Ξ is an approximation to a particular operator, $\Xi_{\rm full}$ say, that does enjoy the property

$$\|\Xi_{\rm full}\,s - s\|_\phi \;=\; 0, \qquad s\in\mathcal{A}. \tag{5.1}$$

A derivation of $\Xi_{\rm full}$ is presented in the next two paragraphs, because it was highly useful to the development of our algorithm. Further, the relation between Ξ and $\Xi_{\rm full}$ may be the main reason for the small number of iterations that are required by the algorithm in many applications, especially when the vector of variables $\underline{x}$ has few components and the multiquadric parameter c is close to zero.

The operator $\Xi_{\rm full}$ from $\mathcal{A}$ to $\mathcal{A}$ can be written in a form that is analogous to expression (4.9). It is defined by the equation

$$(\Xi_{\rm full}\,s)(\underline{x}) \;=\; \sum_{\ell\in\Omega_{\rm all}} \frac{\langle \check{u}_\ell, s\rangle_\phi}{\langle \check{u}_\ell, \check{u}_\ell\rangle_\phi}\,\check{u}_\ell(\underline{x}), \qquad \underline{x}\in\mathcal{R}^d, \quad s\in\mathcal{A}, \tag{5.2}$$

where the functions $\check{u}_\ell$, $\ell\in\Omega_{\rm all}$, are elements of $\mathcal{A}$ without constant terms that are specified below. They are chosen in a way that gives the conditions

$$\langle \check{u}_\ell, \check{u}_\ell\rangle_\phi > 0 \quad \text{and} \quad \langle \check{u}_\ell, \check{u}_m\rangle_\phi = 0, \quad \ell\neq m, \tag{5.3}$$

for all pairs of integers $\ell\neq m$ from $\Omega_{\rm all}$. Therefore the definition (5.2) provides the identities

$$\Xi_{\rm full}\,\check{u}_\ell \;=\; \check{u}_\ell, \qquad \ell\in\Omega_{\rm all}. \tag{5.4}$$

It follows that equation (5.1) holds for all functions that are in the linear subspace of $\mathcal{A}$ that is spanned by $\check{u}_\ell$, $\ell\in\Omega_{\rm all}$, and constant functions, any constant function being admissible as its semi-norm is zero. By using expression (5.3), we deduce that the dimension of this subspace is n, so the subspace is $\mathcal{A}$ itself. Thus the property (5.1) is achieved by $\Xi_{\rm full}$.

For $i=1,2,\ldots,n-1$, we let $\widehat{u}_{\Omega(i)}$ be the function

$$\widehat{u}_{\Omega(i)}(\underline{x}) \;=\; \sum_{k=i}^{n} \Upsilon_{\Omega(i)\,\Omega(k)} \left(\|\underline{x}-\underline{x}_{\Omega(k)}\|^2+c^2 \right)^{1/2} + \text{constant}, \qquad \underline{x}\in\mathcal{R}^d, \tag{5.5}$$

where the coefficients $\Upsilon_{\Omega(i)\,\Omega(k)}$, $k=i,i+1,\ldots,n$, and the constant are defined by the Lagrange conditions

$$\widehat{u}_{\Omega(i)}(\underline{x}_{\Omega(k)}) \;=\; \delta_{ki}, \qquad k=i,i+1,\ldots,n, \tag{5.6}$$

and the constraint $\sum_{k=i}^{n} \Upsilon_{\Omega(i)\,\Omega(k)}=0$. Further, we let $\check{u}_{\Omega(i)}$ be $\widehat{u}_{\Omega(i)}$ without its constant term. Both these functions are familiar if $\mathcal{L}_\ell$ is the set $\{\Omega(k) : k = i,i+1,\ldots,n\}$ in expression (2.4), the value of ℓ being $\Omega(i)$, because in this case equations (2.4), (2.8) and (5.6) imply that $\widehat{u}_{\Omega(i)}$ and $\check{u}_{\Omega(i)}$ are the same as $\widehat{z}_{\Omega(i)}$ and $\check{z}_{\Omega(i)}$, respectively. In general, the first part of expression (5.3) holds for every integer i in $[1,n-1]$, because the conditions (5.6) show that $\check{u}_{\Omega(i)}$ is not a constant function. The other part of expression (5.3) is equivalent to the property $\langle \check{u}_{\Omega(i)}, \check{u}_{\Omega(j)} \rangle_\phi=0$, where i and j are any integers that satisfy $1\le i<j\le n-1$. Then the constraints $\sum_{k=i}^{n} \Upsilon_{\Omega(i)\,\Omega(k)} = \sum_{k=j}^{n} \Upsilon_{\Omega(j)\,\Omega(k)}=0$ with equations (4.5) and (5.5) provide the identity

$$\langle \check{u}_{\Omega(i)}, \check{u}_{\Omega(j)} \rangle_\phi \;=\; \langle \widehat{u}_{\Omega(i)}, \widehat{u}_{\Omega(j)} \rangle_\phi \;=\; -\sum_{k=j}^{n} \Upsilon_{\Omega(j)\,\Omega(k)}\, \widehat{u}_{\Omega(i)}(\underline{x}_{\Omega(k)}). \tag{5.7}$$

It follows from $j>i$ and the values (5.6) that the scalar product (5.7) is zero as required. Therefore the given operator $\Xi_{\rm full}$ does have the property (5.1).

We know now that, if Ξ were the operator $\Xi_{\rm full}$, then the algorithm of Section 2 would terminate no later than the end of the first iteration. Further, if the user of the algorithm provided the value $q=n$ in Step 0, then, for every integer m that occurs, the choice $\mathcal{L}_\ell = \{\Omega(m), \Omega(m+1), \ldots, \Omega(n)\}$ would be made at the end of Step 3, so $\widehat{z}_\ell = \widehat{z}_{\Omega(m)}$ would be the function $\widehat{u}_{\Omega(m)}$, which would imply $\Xi=\Xi_{\rm full}$. On the other hand, because of the work of calculating coefficients of Lagrange functions, the given algorithm can provide a substantial improvement over the best direct method for solving the system (1.4) only if q is much less than n. Therefore the value of q was at most 50 during the development of the algorithm and in the numerical experiments that are reported in the next section. The technique of Step 3, where the elements of $\mathcal{L}_\ell = \mathcal{L}_{\Omega(m)}$ are the integers that minimize the distances $\|\underline{x}_j - \underline{x}_\ell\|$, $j\in\mathcal{L}_\ell$, subject to the condition

$$\mathcal{L}_\ell \subset \{\Omega(m), \Omega(m+1), \ldots, \Omega(n)\}, \tag{5.8}$$

is taken from Beatson and Powell (1994), but the random ordering of the data points in Step 2 is new.

The algorithm has a useful convergence property that depends on the least number ρ that is allowed by the inequality

$$\|s-\Xi s\|_\phi \;\le\; \rho\,\|s\|_\phi, \qquad s\in\mathcal{A}. \tag{5.9}$$

Specifically, the approximation to s^* at the end of the k-th iteration, namely $s^{(k+1)}$, where k is any iteration number, achieves the error bound

$$\|s^*-s^{(k+1)}\|_\phi \;\le\; \rho^k\,\|s^*\|_\phi. \tag{5.10}$$

Indeed, the Krylov subspace feature of the algorithm gives the relation

$$\begin{aligned}\|s^*-s^{(k+1)}\|_\phi \;&=\; \min\Big\{\,\|s^*-\textstyle\sum_{i=1}^k \xi_i\,\Xi^i s^*\|_\phi : \,\xi_i\in\mathcal{R},\;\; i=1,2,\ldots,k\Big\}\\ &\le\; \|(I-\Xi)^k s^*\|_\phi \;\le\; \rho\,\|(I-\Xi)^{k-1}s^*\|_\phi \;\le\; \cdots \;\le\; \rho^k\,\|s^*\|_\phi,\end{aligned} \tag{5.11}$$

where I is the identity operator, and where the last line employs an inductive argument that puts the functions $s=(I-\Xi)^j s^*$, $j=k-1,k-2,\ldots,0$, into inequality (5.9). Therefore, if $\rho<1$ holds, then the algorithm enjoys not only the termination property that is stated in Theorem 1, but also the linear rate of convergence (5.10). Numerical experiments show that $\rho\le\frac{1}{2}$ is usual for moderate values of q, even if the number of interpolation points is very large. Then the linear rate of convergence (5.10) may be crucial to the number of iterations that are required to satisfy the termination condition (2.3).

We show next that ρ is just the spectral radius of $(I-H)$, where H is the $(n-1)\times(n-1)$ symmetric matrix that has the elements

$$H_{ij} \;=\; \langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)}\rangle_\phi \,/\, (\|\check{z}_{\Omega(i)}\|_\phi\,\|\check{z}_{\Omega(j)}\|_\phi), \qquad 1\le i,j\le n-1. \tag{5.12}$$

Let s be a general element of $\mathcal{A}$, written in the form

$$s(\underline{x}) \;=\sum_{j=1}^{n-1} \vartheta_{\Omega(j)}\,\frac{\check{z}_{\Omega(j)}(\underline{x})}{\|\check{z}_{\Omega(j)}\|_\phi} + \text{constant}, \qquad \underline{x}\in\mathcal{R}^d, \tag{5.13}$$

which is analogous to expression (4.12). Equations (5.12) and (5.13) imply that s has the semi-norm

$$\|s\|_\phi \;=\; \langle s,s\rangle_\phi^{1/2} \;=\; (\underline{\vartheta}^T H\,\underline{\vartheta})^{1/2} \;=\; \|H^{1/2}\underline{\vartheta}\|, \tag{5.14}$$

where the components of $\underline{\vartheta}\in\mathcal{R}^{n-1}$ are the coefficients $\vartheta_{\Omega(j)}$, $j=1,2,\ldots,n-1$, and where the vector norm on the right hand side is Euclidean. We deduce from the definition (5.12) that H is positive definite. Moreover, equations (4.9), (5.13) and (5.12) provide the formula

$$\begin{aligned}\Xi s \;\;&=\;\; \sum_{i=1}^{n-1} \frac{\langle \check{z}_{\Omega(i)}, s\rangle_\phi}{\|\check{z}_{\Omega(i)}\|_\phi^2}\,\check{z}_{\Omega(i)} \;\;=\;\; \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} \frac{\vartheta_{\Omega(j)}}{\|\check{z}_{\Omega(j)}\|_\phi}\,\frac{\langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)}\rangle_\phi}{\|\check{z}_{\Omega(i)}\|_\phi^2}\,\check{z}_{\Omega(i)}\\ &=\;\; \sum_{i=1}^{n-1}\Big(\sum_{j=1}^{n-1} H_{ij}\,\vartheta_{\Omega(j)}\Big)\,\frac{\check{z}_{\Omega(i)}}{\|\check{z}_{\Omega(i)}\|_\phi}, \qquad s\in\mathcal{A}.\end{aligned} \tag{5.15}$$

Therefore, if the function $s-\Xi s$ is written in the form (5.13), then it has the vector of coefficients $\underline{\vartheta}-H\underline{\vartheta}$. Hence, corresponding to expression (5.14), the coefficients imply the semi-norm

$$\|s-\Xi s\|_\phi \;=\; \Big((\underline{\vartheta}-H\underline{\vartheta})^T H\,(\underline{\vartheta}-H\underline{\vartheta})\Big)^{1/2} \;=\; \|(I-H)H^{1/2}\underline{\vartheta}\|. \tag{5.16}$$

It follows from equations (5.14) and (5.16) that ρ in condition (5.9) is the least number that satisfies the inequality

$$\|(I-H)H^{1/2}\underline{\vartheta}\| \;\le\; \rho\,\|H^{1/2}\underline{\vartheta}\|, \qquad \underline{\vartheta}\in\mathcal{R}^{n-1}. \tag{5.17}$$

Hence, because of the nonsingularity of $H^{1/2}$, the proof that ρ is the spectral radius of $(I-H)$ is complete.

The definition (5.12) shows that all the diagonal elements of $(I-H)$ are zero. Therefore we deduce from the last two paragraphs that the algorithm has a fast linear rate of convergence, as observed in practice, if the moduli of the normalized scalar products

$$H_{ij} \;=\; \langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)}\rangle_\phi \,/\, (\|\check{z}_{\Omega(i)}\|_\phi\,\|\check{z}_{\Omega(j)}\|_\phi), \qquad 1\le i<j\le n-1, \tag{5.18}$$

are sufficiently small. The following argument goes some way towards explaining the success of values of q from the interval $[10,50]$. Let i and j be integers of expression (5.18), and let δ and Δ be the distances

$$\delta \;=\; \max\Big[\max\{\|\underline{x}_k-\underline{x}_{\Omega(i)}\| : k\in\mathcal{L}_{\Omega(i)}\},\, \max\{\|\underline{x}_\ell-\underline{x}_{\Omega(j)}\| : \ell\in\mathcal{L}_{\Omega(j)}\}\Big] \tag{5.19}$$

and $\Delta=\|\underline{x}_{\Omega(i)}-\underline{x}_{\Omega(j)}\|$. We recall that the sets $\mathcal{L}_{\Omega(i)}$ and $\mathcal{L}_{\Omega(j)}$ have the property that the points $\underline{x}_k$, $k\in\mathcal{L}_{\Omega(i)}$, and $\underline{x}_\ell$, $\ell\in\mathcal{L}_{\Omega(j)}$, are as close as possible to $\underline{x}_{\Omega(i)}$ and $\underline{x}_{\Omega(j)}$, respectively. Therefore, when q is much less than n, the condition $\delta\ll\Delta$ holds for most values of i and j. In this case, an asymptotic expansion is useful for investigating the magnitude of the scalar product $\langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)}\rangle_\phi$. Specifically, the first line of expression (4.5) for the scalar product gives the value

$$\begin{aligned}\langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)}\rangle_\phi \;&=\; -\sum_{k\in\mathcal{L}_{\Omega(i)}}\sum_{\ell\in\mathcal{L}_{\Omega(j)}} \zeta_{\Omega(i)\,k}\,\zeta_{\Omega(j)\,\ell}\,\Big(\|\underline{x}_k-\underline{x}_\ell\|^2+c^2\Big)^{1/2}\\ &=\; -(\Delta^2+c^2)^{1/2}\sum_{k\in\mathcal{L}_{\Omega(i)}}\sum_{\ell\in\mathcal{L}_{\Omega(j)}} \zeta_{\Omega(i)\,k}\,\zeta_{\Omega(j)\,\ell}\left(1+\frac{\|\underline{x}_k-\underline{x}_\ell\|^2-\Delta^2}{\Delta^2+c^2}\right)^{1/2},\end{aligned} \tag{5.20}$$

the difference $\|\underline{x}_k-\underline{x}_\ell\|^2-\Delta^2$ being only of magnitude $\delta\Delta$. Therefore we make the approximation

$$\langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)}\rangle_\phi \;\approx\; -\sum_{k\in\mathcal{L}_{\Omega(i)}}\sum_{\ell\in\mathcal{L}_{\Omega(j)}} \zeta_{\Omega(i)\,k}\,\zeta_{\Omega(j)\,\ell}\left((\Delta^2+c^2)^{1/2}+\frac{\|\underline{x}_k-\underline{x}_\ell\|^2-\Delta^2}{2\,(\Delta^2+c^2)^{1/2}}\right). \tag{5.21}$$

It follows from the constraints $\sum_{k\in\Omega(i)} \zeta_{\Omega(i)\,k} = \sum_{\ell\in\Omega(j)} \zeta_{\Omega(j)\,\ell} = 0$ that this approximation can be written in the form

$$\langle \check{z}_{\Omega(i)}, \check{z}_{\Omega(j)} \rangle_\phi \;\approx\; - \sum_{k\in\mathcal{L}_{\Omega(i)}} \sum_{\ell\in\mathcal{L}_{\Omega(j)}} \zeta_{\Omega(i)\,k}\, \zeta_{\Omega(j)\,\ell}\, \frac{\|(\underline{x}_k - \underline{x}_{\Omega(i)}) - (\underline{x}_\ell - \underline{x}_{\Omega(j)})\|^2}{2\,(\Delta^2 + c^2)^{1/2}}, \quad (5.22)$$

the right hand sides of expressions (5.21) and (5.22) being the same. The last numerator is at most $4\delta^2$, although the numerators of the two previous expressions are $\mathcal{O}(\delta\Delta)$. Thus much cancellation occurs automatically in formula (5.20), which helps to provide a small value of ρ in the error bound (5.10).

It may not be possible to obtain good estimates of the efficiency of the algorithm analytically, because of the interaction between the rate of convergence (5.10) and the conjugate gradient properties that are studied in Section 4. Fewer iterations are required usually if q is increased, because of the remark in the paragraph that includes expression (5.8) that, if $q=n$, then all the scalar products (5.18) are zero, so the first iteration would provide $s^{(2)} = s^*$. Some numerical results of the next section show, however, that sometimes the number of iterations is less for $q=30$ than for $q=50$. Therefore we consider the situation where the amount of preliminary work in Steps 1–4 for $q=50$ would be tolerable, but, because of practical experience with other interpolation problems that are similar to the current one, the decision to set $q=30$ has been made. Then we may retain the set $\mathcal{L}_\ell$ that has been specified already for the function (2.4), but we may seek new values of the coefficients $\zeta_{\ell j}$, $j\in\mathcal{L}_\ell$, the purpose of the change being to provide smaller moduli of the normalized scalar products (5.18), by taking into consideration the positions of some interpolation points whose indices are not in $\mathcal{L}_\ell$.

We study this idea, after making some assumptions that simplify the notation without losing generality. We let ℓ have the value $\ell=\Omega(1)=1$, we let $\mathcal{L}_\ell$ be the set $\{1,2,\ldots,q\}$, and we allow the coefficients $\zeta_{\ell j}$, $j=1,2,\ldots,q$, to depend on the positions of all the data points $\underline{x}_i$, $i=1,2,\ldots,n$. Then the off-diagonal matrix elements H_{1j}, $j=2,3,\ldots,n-1$, are close to zero as required, if $\check{z}_1 = \check{z}_{\Omega(1)}$ is nearly orthogonal to all functions in the linear space

$$\mathcal{B} \;=\; \text{span}\left\{ \check{z}_{\Omega(j)} \,:\, j=2,3,\ldots,n-1 \right\}. \quad (5.23)$$

We let s be a general element of $\mathcal{B}$ for the remainder of this section. Therefore, because $\Omega(1)=1$ is excluded from the sets $\mathcal{L}_j$, $j=2,3,\ldots,n-1$, it has the form

$$s(\underline{x}) \;=\; \sum_{j=2}^{n} \sigma_j \left(\|\underline{x} - \underline{x}_j\|^2 + c^2 \right)^{1/2}, \qquad \underline{x}\in\mathcal{R}^d, \quad (5.24)$$

where the coefficients σ_j satisfy $\sum_{j=2}^n \sigma_j = 0$. On the other hand, $\check{z}_1$ has the form

$$\check{z}_1(\underline{x}) \;=\; \sum_{j=1}^{q} \zeta_{1j} \left(\|\underline{x} - \underline{x}_j\|^2 + c^2 \right)^{1/2}, \qquad \underline{x}\in\mathcal{R}^d, \quad (5.25)$$

and we are seeking optimal values of the coefficients ζ_{1j}, $j=1,2,\ldots,q$, subject to the constraints $\zeta_{11}\neq 0$ and $\sum_{j=1}^{q}\zeta_{1j}=0$. For any trial choice of $\check{z}_1$, we let $V(\check{z}_1)$ be the largest normalized scalar product

$$V(\check{z}_1) \;=\; \max\Big\{\, |\langle \check{z}_1, s\rangle_\phi| \,/\, (\|\check{z}_1\|_\phi \,\|s\|_\phi) : \; s\in\mathcal{B},\; \|s\|_\phi\neq 0 \Big\}. \qquad (5.26)$$

Further, following a suggestion in Powell (2002), we say that the coefficients of $\check{z}_1$ are optimal if the resultant value of $V(\check{z}_1)$ is as small as possible. The use of such coefficients, however, is not an improvement over the present algorithm, because we discovered recently that already the coefficients of the algorithm provide the least value of $V(\check{z}_1)$. A proof of this property, using the assumptions and notation of this paragraph, is given below.

Theorem 2: Let the coefficients of the function (2.4) be defined by Step 4 of the algorithm in the case $\ell=1$ and $\mathcal{L}_\ell=\{1,2,\ldots,q\}$. Further, let $\check{z}_1$ be $\widehat{z}_1$ without its constant term. This choice of $\check{z}_1$ provides the least value of expression (5.26), when $\check{z}_1$ is allowed to be any function of the form (5.25) whose coefficients satisfy $\zeta_{11}\neq 0$ and $\sum_{j=1}^{q}\zeta_{1j}=0$.

Proof: Let $\check{z}_1$ be any function of the form (5.25), where $\zeta_{11}\neq 0$ and $\sum_{j=1}^{q}\zeta_{1j}=0$. Then $\check{z}_1$ is not constant, and, because of the nonsingularity of the system (1.4), the values $\check{z}_1(\underline{x}_i)$, $i=1,2,\ldots,n$, are not all the same, but $\check{z}_1(\underline{x}_i)=\eta$, $i=2,3,\ldots,n$, is possible for some real number η. Then equations (4.5) and (5.24) would give $\langle \check{z}_1, s\rangle_\phi=-\eta\sum_{j=2}^{n}\sigma_j=0$ for every function of the form (5.24) whose coefficients sum to zero, so $V(\check{z}_1)$ would be zero. Further, the algorithm would calculate the function

$$\widehat{z}_1(\underline{x}) \;=\; [\,\check{z}_1(\underline{x})-\eta\,]\,/\,[\,\check{z}_1(\underline{x}_1)-\eta\,], \qquad \underline{x}\in\mathcal{R}^d, \qquad (5.27)$$

because it has the form (2.4) and achieves the Lagrange conditions (2.8). Thus $\check{z}_1$ in the statement of the theorem would be the function $\check{z}_1$ that is under consideration multiplied by the constant $1/[\check{z}_1(\underline{x}_1)-\eta]$. This change to $\check{z}_1$ would preserve the zero value of $V(\check{z}_1)$, so the theorem holds in the present special case.

Otherwise, when $\check{z}_1$ is a general function of the form under consideration, and the values $\check{z}_1(\underline{x}_i)$, $i=2,3,\ldots,n$, are not all the same, we seek a function $\check{t}\in\mathcal{B}$ such that the maximum on the right hand side of expression (5.26) is attained by $s=\check{t}$. We let $\widehat{t}$ be the function

$$\widehat{t}(\underline{x}) \;=\; \sum_{j=2}^{n}\check{\tau}_j\left(\|\underline{x}-\underline{x}_j\|^2+c^2\right)^{1/2}+\text{constant}, \qquad \underline{x}\in\mathcal{R}^d, \qquad (5.28)$$

whose parameters are defined by the conditions $\widehat{t}(\underline{x}_i)=\check{z}_1(\underline{x}_i)$, $i=2,3,\ldots,n$, and $\sum_{j=2}^{n}\check{\tau}_j=0$, and we let $\check{t}$ be $\widehat{t}$ without its constant term, which is an element of $\mathcal{B}$. It follows from expression (4.5) for the scalar product that the general element (5.24) of $\mathcal{B}$ has the property

$$\begin{aligned}\langle s,\check{t}\rangle_\phi \;&=\; -\textstyle\sum_{i=2}^{n}\sigma_i\,\check{t}(\underline{x}_i) \;=\; -\sum_{i=2}^{n}\sigma_i\,\widehat{t}(\underline{x}_i)\\ &=\; -\textstyle\sum_{i=2}^{n}\sigma_i\,\check{z}_1(\underline{x}_i) \;=\; \langle s,\check{z}_1\rangle_\phi, \qquad s\in\mathcal{B}. \qquad (5.29)\end{aligned}$$

Therefore the Cauchy–Schwarz inequality provides the relation

$$|\langle s,\check{z}_1\rangle_\phi| \;=\; |\langle s,\check{t}\rangle_\phi| \;\le\; \|s\|_\phi\,\|\check{t}\|_\phi \;=\; \|s\|_\phi\,\langle \check{t},\check{z}_1\rangle_\phi \,/\, \|\check{t}\|_\phi, \tag{5.30}$$

the last equation being true because $s=\check{t}$ is admissible in the identity (5.29). Further, $\|\check{t}\|_\phi$ is positive because $\check{t}(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, is not constant and, similarly, $\|\check{z}_1\|_\phi$ is also positive. Thus the relation (5.30) is equivalent to the condition

$$|\langle \check{z}_1,\check{t}\rangle_\phi| \,/\, (\|\check{z}_1\|_\phi\,\|\check{t}\|_\phi) \;\ge\; |\langle \check{z}_1,s\rangle_\phi| \,/\, (\|\check{z}_1\|_\phi\,\|s\|_\phi), \qquad s\in\mathcal{B},\;\; \|s\|_\phi\neq 0, \tag{5.31}$$

which shows that the maximum of expression (5.26) is achieved when $s=\check{t}$.

We have to establish the bound $V(\check{z}_1)\ge V(\check{z}_1^*)$, for all the functions $\check{z}_1$ that are mentioned in the statement of the theorem, $\check{z}_1^*$ being the particular choice of $\check{z}_1$ that is stated. Let $\check{t}$ be as in the previous paragraph when $\check{z}_1$ is general, and let $\check{t}^*$ be the function $\check{t}$ that occurs in the case $\check{z}_1=\check{z}_1^*$. It follows from condition (5.31) that it is sufficient to prove the inequality

$$|\langle \check{z}_1,\check{t}\rangle_\phi| \,/\, (\|\check{z}_1\|_\phi\,\|\check{t}\|_\phi) \;\ge\; |\langle \check{z}_1^*,\check{t}^*\rangle_\phi| \,/\, (\|\check{z}_1^*\|_\phi\,\|\check{t}^*\|_\phi), \tag{5.32}$$

which is equivalent to the bound

$$\|\check{t}\|_\phi \,/\, \|\check{z}_1\|_\phi \;\ge\; \|\check{t}^*\|_\phi \,/\, \|\check{z}_1^*\|_\phi, \tag{5.33}$$

because of the last part of the relation (5.30). Now, if $\check{z}_1$ is multiplied by a nonzero constant, then $\check{t}$ is multiplied by the same constant, so the left hand side of this bound is unchanged. Therefore, because the coefficients of the term $(\|\underline{x}-\underline{x}_1\|^2+c^2)^{1/2}$, $\underline{x}\in\mathcal{R}^d$, in both $\check{z}_1$ and $\check{z}_1^*$ are nonzero, we assume without loss of generality that they are the same. In this case, $\check{z}_1$ has the form

$$\check{z}_1(\underline{x}) \;=\; \check{z}_1^*(\underline{x}) + \sum_{j=2}^{q} \check{\sigma}_j \left(\|\underline{x}-\underline{x}_j\|^2+c^2\right)^{1/2}, \qquad \underline{x}\in\mathcal{R}^d, \tag{5.34}$$

for some coefficients $\check{\sigma}_j$, $j=2,3,\ldots,q$, that sum to zero. Thus the difference $\check{z}_1-\check{z}_1^*=\check{s}$, say, is an element of $\mathcal{B}$. Moreover, because $\check{z}_1^*$ is $\widehat{z}_1^*$ without its constant term, where $\widehat{z}_1^*$ is defined by the Lagrange conditions $\widehat{z}_1^*(\underline{x}_j)=\delta_{j1}$, $j=1,2,\ldots,q$, the values $\check{z}_1^*(\underline{x}_j)$, $j=2,3,\ldots,q$, are all the same. Hence $\check{s}$ satisfies the orthogonality condition

$$\langle \check{s},\check{z}_1^*\rangle_\phi \;=\; -\sum_{j=2}^{q} \check{\sigma}_j\, \check{z}_1^*(\underline{x}_j)=0, \tag{5.35}$$

which implies the equation

$$\|\check{z}_1\|_\phi^2 \;=\; \|\check{z}_1^*+\check{s}\|_\phi^2 \;=\; \|\check{z}_1^*\|_\phi^2 + \|\check{s}\|_\phi^2. \tag{5.36}$$

Now $\check{t}$ is such that the values $(\check{z}_1-\check{t})(\underline{x}_i)$, $i=2,3,\ldots,n$, are all the same, and the definition of $\check{s}$ gives the identities

$$\check{z}_1(\underline{x}_i) - \check{t}(\underline{x}_i) \;=\; \check{z}_1^*(\underline{x}_i) - (\check{t}-\check{s})(\underline{x}_i), \qquad i=2,3,\ldots,n. \tag{5.37}$$

Thus, because $\check{t}-\check{s}$ is in $\mathcal{B}$ and because $\check{t}^*$ is the unique element of $\mathcal{B}$ such that the values $(\check{z}_1^*-\check{t}^*)(\underline{x}_i)$, $i=2,3,\ldots,n$, are independent of i, we deduce $\check{t}^*=\check{t}-\check{s}$. Moreover, by putting $s=\check{s}$ into the relation (5.29) in the case $\check{z}_1=\check{z}_1^*$ and $\check{t}=\check{t}^*$, we obtain $\langle\check{s},\check{t}^*\rangle_\phi=\langle\check{s},\check{z}_1^*\rangle_\phi$. These remarks and the property (5.35) provide the equation

$$\|\check{t}\|_\phi^2 \;=\; \|\check{t}^*+\check{s}\|_\phi^2 \;=\; \|\check{t}^*\|_\phi^2+\|\check{s}\|_\phi^2. \tag{5.38}$$

It follows from expressions (5.33), (5.38) and (5.36) that we are trying to establish the inequality

$$\Big(\|\check{t}^*\|_\phi^2+\|\check{s}\|_\phi^2\Big)\Big/\Big(\|\check{z}_1^*\|_\phi^2+\|\check{s}\|_\phi^2\Big) \;\geq\; \|\check{t}^*\|_\phi^2\Big/\|\check{z}_1^*\|_\phi^2. \tag{5.39}$$

This condition holds if $\|\check{s}\|_\phi$ is zero, and otherwise it is equivalent to $\|\check{t}^*\|_\phi\le\|\check{z}_1^*\|_\phi$. We recall that the right hand sides of expressions (5.32) and (5.33) are the same. Because the first of them is at most one by the Cauchy–Schwarz inequality, the bound $\|\check{t}^*\|_\phi/\|\check{z}_1^*\|_\phi\le 1$ is satisfied. Therefore the theorem is true. □

6. Numerical results and discussion

The first set of numerical results of this section compares our algorithm with the method that is described by Powell (1997) for thin plate spline interpolation. Therefore we employ the four test problems of that paper, which are as follows.

In every case there are only two variables, the positions of the data points $\underline{x}_i\in\mathcal{R}^2$, $i=1,2,\ldots,n$, are generated randomly, and the right hand sides f_i, $i=1,2,\ldots,n$, of the interpolation equations (1.1) are random numbers from the distribution that is uniform on $[-1,1]$. Further, the parameter ε of the termination condition (2.3) is given the value $\varepsilon=10^{-10}$. In Problem A, each data point is chosen independently from the uniform distribution on the unit disc $\{\underline{x}:\|\underline{x}\|_2\le 1\}$. In Problem B, the data points have the form

$$\underline{x}_i \;=\; \Big(|\log\|\widehat{\underline{x}}_i\|^2|\,/\,\|\widehat{\underline{x}}_i\|^2\Big)^{1/2}\,\widehat{\underline{x}}_i, \qquad i=1,2,\ldots,n, \tag{6.1}$$

where each $\widehat{\underline{x}}_i$ is sampled independently from the uniform distribution on the unit disc. Thus some of the interpolation points can be far from the origin. In the other two problems, the data points lie on N straight line segments, where $N=2$ and $N=10$ occur in Problems C and D, respectively. The end points of each line segment are picked randomly from the uniform distribution on the unit square $[0,1]\times[0,1]$, except that, if the length of a line segment is less than one, then both end points are replaced by new ones, generated as before until they are acceptable.

Problem	n	$q=10$	$q=20$	$q=30$	$q=40$	$q=50$
A	200	14–15	9–10	7–8	6–7	6–6
A	500	17–18	10–11	8–9	7–8	6–7
A	1000	19–21	11–12	9–10	8–9	7–8
A	2000	21–22	12–13	10–10	9–9	8–8
A	5000	23–26	14–14	11–12	10–10	9–9
A	10000	26–28	15–15	11–12	10–11	9–10
B	200	16–18	10–10	8–8	7–8	6–7
B	500	18–22	11–13	9–10	7–9	7–8
B	1000	20–23	12–13	10–11	8–10	7–8
B	2000	22–27	13–14	10–12	8–10	8–9
B	5000	25–28	14–15	11–12	9–10	8–9
B	10000	28–33	16–17	12–12	10–11	8–10
C	200	8–11	7–9	6–8	5–7	5–6
C	500	10–11	8–9	6–8	6–8	6–7
C	1000	10–14	8–9	7–8	6–8	6–8
C	2000	10–13	8–10	7–9	7–9	6–8
C	5000	11–13	8–10	8–9	7–9	6–9
C	10000	11–19	8–13	7–12	7–10	7–10
D	200	15–16	10–11	8–9	7–8	6–8
D	500	17–20	11–12	10–11	8–10	8–9
D	1000	19–21	12–14	11–12	9–10	9–10
D	2000	21–23	13–15	12–13	10–12	10–11
D	5000	22–25	15–17	13–14	11–13	11–12
D	10000	23–27	15–20	13–16	12–15	10–14

Table 1: Iteration counts for four problems with $c=0$ and $d=2$

Further, n/N data points are placed on each line segment, n being a multiple of 10. Their positions are chosen randomly and independently from the distribution that is uniform on the line segment.

Table 1 shows the values of n and q that were tried. The parameter c of the multiquadric form (1.2) was set to zero. Ten interpolation calculations were solved for each problem and for each choice of n and q, these calculations being different because of the random numbers that were used, not only to specify the data, but also in Step 2 of the algorithm. In every case the number of iterations was recorded, the least and greatest of these numbers for each set of ten calculations being reported in the main part of Table 1. The narrowness of each range shows that the amount of computation of the algorithm is insensitive to the randomness in the data. A nonpositive Cholesky pivot, mentioned near the end of Section 3, occurred in a Problem D experiment with $n = 10000$. That case was replaced by another one with different random numbers.

The most welcome feature of Table 1 is that, throughout the experiments, the number of iterations is much less than n, although the termination condition requires ten decimal places of accuracy. Further, only a little growth occurs in the numbers of iterations when n is increased. We conclude that, when c is zero, the given iterative procedure is highly successful for interpolation to values of functions of only two variables. Most of the iteration counts of Table 1 are about 25% more than the corresponding results of Powell (1997) for thin plate spline interpolation, except that our iteration counts are lower for $q = 10$, which may be because the earlier algorithm does not include a Krylov subspace technique. Problem C requires fewest iterations in the table, as it is relatively easy to construct suitable approximations to Lagrange functions when the interpolation points are confined to two line segments in $\mathcal{R}^2$.

The last remark suggests that, if the number of variables d is large, but if all the data points $\underline{x}_i$, $i = 1, 2, \ldots, n$, are on a smooth two dimensional manifold of $\mathcal{R}^d$, then the number of iterations of the algorithm may depend on n and q as in Table 1. Excellent support for this suggestion can be found in Table 6 of Powell (2002). For example, when $n = 1024$, then the numbers of iterations are 20–22 and 6–8 for $q = 10$ and $q = 50$, respectively, for every value of d in the set $\{5, 10, 20, 40\}$. Instead of investigating data on manifolds, however, which is an important subject for many applications, we consider the difficult generalization of Problem A that allows d to be any positive integer. Specifically, each of the interpolation points $\underline{x}_i$, $i = 1, 2, \ldots, n$, is sampled independently from the distribution that is uniform throughout the ball $\{\underline{x} : \|\underline{x}\|_2 \leq 1\} \subset \mathcal{R}^d$. Further, we retain the right hand sides $f_i \in [-1, 1]$, $i = 1, 2, \ldots, n$, and the value $\varepsilon = 10^{-10}$ from the second paragraph of this section, we take n and q from Table 1 without the extreme values of n, we continue with the choice $c = 0$ of the multiquadric parameter, we let the number of variables d be 5, 10, 20 and 40, and we generate ten versions of each interpolation calculation by using different random numbers as before. We omit $n = 200$ because its results are uninteresting, and we avoid $n = 10000$ because, due to the storage limitations of the computer that was used and the high number of iterations, the amount of work would be intolerable.

The iteration counts of the algorithm for the larger values of d are displayed in Table 2, the pairs of numbers being as in Table 1. We expect the counts to be much greater than the Problem A entries of Table 1, because, for the given n and q with $d \geq 5$, the use of functions with only q multiquadric terms to approximate Lagrange functions with up to n centres is very crude. Therefore the speed of convergence of the iterative procedure is now strongly dependent on the efficiency of the Krylov subspace technique. A remarkable triumph for the technique is that nearly all the $d = 40$ results in Table 2 are less than the corresponding $d = 20$ results. One difference between $d = 20$ and $d = 40$ is that, in the latter case,

d	n	$q=10$	$q=20$	$q=30$	$q=40$	$q=50$
5	500	45–47	32–34	26–27	20–21	17–18
5	1000	61–64	43–45	34–36	27–28	22–23
5	2000	81–84	58–61	44–46	35–37	29–30
5	5000	128–130	82–87	64–69	50–52	41–42
10	500	50–54	47–50	42–46	37–40	34–36
10	1000	71–76	63–70	59–64	56–61	49–53
10	2000	104–109	91–99	88–92	79–88	74–79
10	5000	173–180	149–156	144–148	133–146	124–134
20	500	47–49	43–45	45–49	45–50	47–49
20	1000	66–70	56–60	59–65	59–71	62–70
20	2000	90–94	79–85	84–91	86–94	89–96
20	5000	145–152	120–129	128–136	137–150	143–171
40	500	48–52	41–43	42–46	44–45	45–48
40	1000	63–67	50–53	52–55	55–60	57–62
40	2000	77–81	63–66	64–67	68–70	70–79
40	5000	113–119	88–91	87–91	91–102	97–110

Table 2: Iteration counts for Problem A with $c=0$ and $d\geq 5$

tiny distances between interpolation points are less likely to occur. Therefore, in another experiment, the points $\underline{x}_i$, $i=1,2,\ldots,n$, were picked randomly in sequence from the unit ball $\{\underline{x} : \|\underline{x}\|_2 \leq 1\}$, but the trial position of $\underline{x}_i$, $i\geq 2$, was accepted if and only if it satisfied the constraints

$$\|\underline{x}_i - \underline{x}_j\| \;\geq\; (1/n)^{1/d}, \qquad j=1,2,\ldots,i-1. \tag{6.2}$$

Otherwise, the construction of $\underline{x}_i$ was restarted. Thus unusually small distances between points were suppressed. We found, however, that the changes to the iteration counts due to this modification are negligible, so the relatively good performance of the algorithm for $d=40$ is unexplained. Another interesting feature of Table 2, which provided encouragement for the analysis at the end of Section 5, is that, for $d\geq 20$, the number of iterations when $q=50$ is usually greater than the corresponding number when q is 20, 30 or 40. Therefore making a good choice of q in these cases requires some care.

Positive values of the multiquadric parameter c are employed in many applications (Hardy, 1990), and often c has the magnitude of nearest neighbour distances between the data points $\underline{x}_i$, $i=1,2,\ldots,n$. In the experiments of Table 2, this magnitude is approximately the right hand side of inequality (6.2), because the sum of the volumes of n balls of radius $(1/n)^{1/d}$ in $\mathcal{R}^d$ is exactly the volume of the region $\{\underline{x} : \|\underline{x}\|_2 \leq 1\}$ that contains the interpolation points. Therefore most

d	n	$c=(1/n)^{1/d}$			$c=2\,(1/n)^{1/d}$		
		$q=10$	$q=30$	$q=50$	$q=10$	$q=30$	$q=50$
2	500	22.6	9.3	7.4	37.8	12.8	9.0
2	1000	25.0	9.9	8.0	43.2	13.3	9.1
2	2000	27.8	10.9	9.0	48.8	13.8	10.1
2	5000	32.1	12.0	9.9	56.3	15.2	11.0
5	500	86.5	33.2	22.0	209.1	61.9	39.7
5	1000	122.1	42.6	27.9	329.7	85.5	53.7
5	2000	179.0	57.8	36.2	514.8	123.7	76.4
5	5000	292.9	89.2	52.8	911.9	214.2	127.3
10	500	118.8	80.7	45.7	307.4	149.1	82.4
10	1000	178.1	129.6	69.2	497.7	259.3	138.3
10	2000	269.2	208.6	111.0	844.7	467.8	235.0
10	5000	492.4	387.0	211.6	1707.7	1051.7	469.9
20	500	91.4	120.3	94.1	175.9	355.8	171.8
20	1000	136.6	190.0	163.0	278.3	674.5	361.0
20	2000	206.3	301.0	277.7	434.7	1231.8	745.9
20	5000	349.0	500.9	531.5	811.3	2468.7	1797.2
40	500	72.1	87.6	107.9	105.7	187.5	310.5
40	1000	100.7	120.5	164.8	164.3	293.5	632.4
40	2000	147.5	167.0	249.1	258.6	458.8	1220.1
40	5000	239.1	275.2	421.8	424.3	826.0	2634.8

Table 3: Average iteration counts for Problem A with $c>0$

of the Problem A calculations of Tables 1 and 2 were rerun, using the values of c that are given at the top of Table 3 instead of $c=0$. As before, ten different calculations were generated for each choice of d, n, q and c by using different random numbers. The entries in the main part of Table 3 are the averages of the iteration counts that occurred.

We see in Table 3 that, for $d=2$, the growth in the number of iterations remains small when n is increased, but this highly useful property is not inherited by large values of d. In particular, when $d=40$ and $q=50$ on the right hand side of the table, the number of iterations seems to be nearly proportional to n. Furthermore, in contrast to Table 2, we find now that the intermediate choice $q=30$ is never attractive, this value being particularly bad in the $d=20$ results. It is also surprising that $q=10$ is the clear winner over $q=30$ and $q=50$ for the largest d. Therefore $q=5$ was tried also in the $d=40$ calculations on the right hand side of the table, the average numbers of iterations being 157.2, 253.8, 419.3

n	q	Preliminary work	Each iteration	Problem A with $c=0$	Problem A with $c>0$	Direct method
500	10	0.17	0.030	0.70	1.32	2.84
500	30	0.68	0.033	0.94	1.10	2.84
500	50	2.05	0.035	2.29	2.35	2.84
1000	10	0.64	0.126	3.09	5.94	23.0
1000	30	1.75	0.132	2.94	3.53	23.0
1000	50	4.80	0.143	5.89	6.13	23.0
2000	10	2.51	0.535	14.2	29.0	196.2
2000	30	4.96	0.553	10.4	12.7	196.2
2000	50	11.60	0.596	16.4	17.6	196.2
5000	10	15.3	3.32	96.6	202.1	3183.8
5000	30	22.4	3.37	59.4	74.3	3183.8
5000	50	39.9	3.49	71.3	78.3	3183.8

Table 4: Timings in seconds on an Ultra 10 workstation with $d=2$

and 737.1 for the given values of n. These remarks confirm that the choice of q should depend on the application. Moreover, the iteration counts in Table 3 can be halved by reducing the tolerance parameter ε from 10^{-10} to 10^{-5}, which is appropriate for most practical calculations with $d\geq 10$ and $n\leq 5000$.

Finally, we compare the amount of computation of our algorithm with that of the direct method of the last paragraph of Section 3. If the number of iterations is prescribed, then the only contribution from the dimension d to the amount of work is in the calculation of the matrix elements

$$\Phi_{ij} \;=\; (\|\underline{x}_i-\underline{x}_j\|^2+c^2)^{1/2}, \qquad 1\leq i\leq j\leq n. \tag{6.3}$$

Therefore we restrict attention to only one value of d, namely $d=2$, in the case when all the terms (6.3) are stored initially, because they are required on every iteration. Some timings in seconds for our algorithm and the direct method are given in Table 4. All the calculations were programmed in Fortran 77, and run on a Sun Ultra 10 workstation, the times being measured by the `DTIME` instruction of the Sun Fortran Library. Separate measurements of the preliminary work and of a single iteration are reported in Table 4, the values of n and q being taken from Table 3. The total times when the iterative algorithm is applied to two versions of Problem A are shown too, the positive value of c being $2\,(1/n)^{1/d}$ (see Tables 1 and 3 for the numbers of iterations). We find that these times compare favourably with the measurements for the direct method in the last column of the table. The running time of each iteration is insensitive to q, because it is dominated by the matrix vector product in Step 7 of Section 2. This task is very onerous if the matrix elements (6.3) are recalculated whenever they are required, and then the dependence of the number of iterations on the choice of q becomes more important.

The Fortran 77 implementation of the given algorithm is available free of charge from the third author, at the e-mail address mjdp@cam.ac.uk. It has been run successfully on interpolation calculations that have up to 10000 data points, although the fast storage of our Ultra 10 workstation cannot hold all the elements (6.3) in the largest calculations. The software should be useful to many applications, because the multiquadric method is particularly suitable for multivariable interpolation when the positions of the data points are general. Indeed, the large values of d that occur in data mining do not introduce difficulties, and the given algorithm takes advantage automatically of the situation when the data points happen to lie on a low dimensional manifold in $\mathcal{R}^d$.

References

R.K. Beatson and M.J.D. Powell (1994), "An iterative method for thin plate spline interpolation that employs approximations to Lagrange functions", in *Numerical Analysis 1993*, eds. D.F. Griffiths and G.A. Watson, Longman Scientific & Technical (Burnt Mill), pp. 17–39.

J.B. Cherrie, R.K. Beatson and G.N. Newsam (2002), "Fast evaluation of radial basis functions: methods for generalised multiquadrics in $\mathcal{R}^n$", *SIAM J. Sci. Statist. Comput.*, Vol. 23, pp. 1549–1571.

A.C. Faul (2001), *Iterative Techniques for Radial Basis Function Interpolation*, Ph.D. Dissertation, University of Cambridge.

A.C. Faul and M.J.D. Powell (2000), "Krylov subspace methods for radial basis function interpolation", in *Numerical Analysis 1999*, eds. D.F. Griffiths and G.A. Watson, Chapman & Hall (London), pp. 115–141.

R.L. Hardy (1990), "Theory and applications of the multiquadric-biharmonic method", *Comput. Math. Applic.*, Vol. 19, pp. 163–208.

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

M.J.D. Powell (1997), "A new iterative method for thin plate spline interpolation in two dimensions", *Annals of Numerical Mathematics*, Vol. 4, pp. 519–527.

M.J.D. Powell (2002), "Radial basis function methods for interpolation to functions of many variables", *HERMIS mu pi*, Vol. 3, pp. 1–24.

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