DAMTP 2001/NA11

Radial basis function methods for interpolation to functions of many variables[1]

M.J.D. Powell

Abstract: A review of interpolation to values of a function $f(\underline{x})$, $\underline{x} \in \mathcal{R}^d$, by radial basis function methods is given. It addresses the nonsingularity of the interpolation equations, the inclusion of polynomial terms, and the accuracy of the approximation $s \approx f$, where s is the interpolant. Then some numerical experiments investigate the situation when the data points are on a low dimensional nonlinear manifold in $\mathcal{R}^d$. They suggest that the number of data points that are necessary for good accuracy on the manifold is independent of d, even if d is very large. The experiments employ linear and multiquadric radial functions, because an iterative algorithm for these cases was developed at Cambridge recently. The algorithm is described briefly. Fortunately, the number of iterations is small when the data points are on a low dimensional manifold. We expect these findings to be useful, because the manifold setting for large d is similar to typical distributions of interpolation points in data mining applications.

Department of Applied Mathematics and Theoretical Physics,
University of Cambridge,
Silver Street,
Cambridge CB3 9EW,
England.

October, 2001 (revised July, 2002).

[1]presented at the Fifth Hellenic-European Conference on Computer Mathematics and its Applications (Athens, September, 2001)

1. Introduction

Let $f(\underline{x})$, $\underline{x} \in \mathcal{R}^d$, be a real function of d variables, and let the values $f(\underline{x}_i)$, $i=1,2,\ldots,n$, be given, where the points $\underline{x}_i$, $i=1,2,\ldots,n$, are all different. The interpolation problem is to construct a function $s(\underline{x})$, $\underline{x} \in \mathcal{R}^d$, that satisfies the conditions

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

The usual approach is to pick an n-dimensional linear space $\mathcal{A}$, say, of real functions from $\mathcal{R}^d$ to $\mathcal{R}$, that is spanned by a convenient basis $a_j \in \mathcal{A}$, $j=1,2,\ldots,n$, and to seek an interpolant of the form

$$s(\underline{x}) = \sum_{j=1}^{n} \lambda_j \, a_j(\underline{x}), \qquad \underline{x} \in \mathcal{R}^d. \tag{1.2}$$

Then the conditions (1.1) provide the linear system of equations

$$\sum_{j=1}^{n} \lambda_j \, a_j(\underline{x}_i) = f(\underline{x}_i), \qquad i=1,2,\ldots,n, \tag{1.3}$$

which defines the coefficients λ_j, $j=1,2,\ldots,n$, uniquely, provided that the matrix of the system is nonsingular. We see that this matrix is the $n\times n$ matrix Φ that has the elements $\Phi_{ij}=a_j(\underline{x}_i)$, $1\le i,j\le n$.

If $d\ge 2$, however, if all the basis functions are continuous, and if the choice of $\mathcal{A}$ does not depend on the interpolation points $\underline{x}_i$, $i=1,2,\ldots,n$, then positions of these points exist such that Φ is singular. This remark is well known, because there is room in two or more dimensions to exchange the positions of two of the interpolation points in a continuous way that keeps the points apart, and then the corresponding value of $\det \Phi$ must pass through zero. It is therefore very useful that many radial basis function methods for interpolation employ a linear space $\mathcal{A}$ that guarantees that Φ is nonsingular. In the multiquadric case, for example, $\mathcal{A}$ is the n-dimensional linear space that is spanned by the basis functions

$$a_j(\underline{x}) = \sqrt{\|\underline{x}-\underline{x}_j\|^2 + c^2}, \qquad \underline{x} \in \mathcal{R}^d, \quad j=1,2,\ldots,n, \tag{1.4}$$

where the vector norm is Euclidean, and where c is a positive constant. Many applications of this method to real problems are surveyed by Hardy (1990), and they began before the nonsingularity of Φ had been proved (Micchelli, 1986).

The general form of radial basis function interpolation without a constant term is as follows. A real function $\phi(r)$, $r\ge 0$, is chosen, which takes the values $\phi(r)=(r^2+c^2)^{1/2}$ in the multiquadric case that has been mentioned. It provides the basis functions

$$a_j(\underline{x}) = \phi(\|\underline{x}-\underline{x}_j\|), \qquad \underline{x} \in \mathcal{R}^d, \quad j=1,2,\ldots,n, \tag{1.5}$$

as in equation (1.4). Other choices of ϕ include the linear case $\phi(r)=r$, $r\geq 0$, the Gaussian case $\phi(r)=\exp(-cr^2)$, $r\geq 0$, and the inverse multiquadric case $\phi(r)=(r^2{+}c^2)^{-1/2}$, $r\geq 0$, where c is still a positive constant. For each of them, the matrix Φ of the linear system (1.3) is nonsingular if $n\geq 2$, and if the data points $\underline{x}_i$, $i=1,2,\ldots,n$, are all different. On the other hand, the choice $\phi(r)=r^2$, $r\geq 0$, would be unsuitable, because then all of the functions (1.5) would be quadratic polynomials, so Φ would be singular if n were greater than the dimension of the linear space of polynomials of degree at most two from $\mathcal{R}^d$ to $\mathcal{R}$, this dimension being $\frac{1}{2}(d{+}1)(d{+}2)$. We are going to study some examples with $n\geq 1000$, because large values of n occur in many practical calculations.

We write the system (1.3) in the matrix form

$$\Phi\,\underline{\lambda}=\underline{f}, \tag{1.6}$$

where $\underline{\lambda}$ and $\underline{f}$ are the vectors in $\mathcal{R}^n$ that have the components λ_j, $j=1,2,\ldots,n$, and $f(\underline{x}_i)$, $i=1,2,\ldots,n$. Equation (1.5) shows that Φ has the elements

$$\Phi_{ij}=\phi(\|\underline{x}_i-\underline{x}_j\|), \qquad 1\leq i,j\leq n. \tag{1.7}$$

Therefore Φ is a symmetric matrix. Further, Φ is positive definite in the Gaussian and inverse multiquadric cases. The proof of nonsingularity in the multiquadric and linear cases includes a highly useful intermediate result. It is that, if $\underline{v}$ is any nonzero vector in $\mathcal{R}^n$ whose components sum to zero, then $\underline{v}^T\Phi\,\underline{v}$ is negative. It follows that Φ has $n-1$ negative eigenvalues and one positive eigenvalue, which implies nonsingularity (Micchelli, 1986).

In many applications, the linear space $\mathcal{A}$ should include constant functions, but the choices of ϕ above do not give this property when n is finite. Therefore an augmented version of the radial basis function method may be preferred, where the interpolant s has the form

$$s(\underline{x})=\sum_{j=1}^{n}\lambda_j\,\phi(\|\underline{x}-\underline{x}_j\|)+\alpha, \qquad \underline{x}\in\mathcal{R}^d. \tag{1.8}$$

Here α is another real parameter that provides an arbitrary constant term explicitly. In this version the constraint

$$\sum_{j=1}^{n}\lambda_j=0 \tag{1.9}$$

is imposed on the original parameters, in order that the available functions s are still the elements of an n-dimensional linear space. The conditions (1.1) and (1.9) show that the parameters of the interpolant (1.8) have to satisfy the $(n{+}1)\times(n{+}1)$ system of linear equations

$$\left(\begin{array}{c|c}\Phi & \underline{e}\\ \hline \underline{e}^T & 0\end{array}\right)\left(\begin{array}{c}\underline{\lambda}\\ \hline \alpha\end{array}\right)=\left(\begin{array}{c}\underline{f}\\ \hline 0\end{array}\right), \tag{1.10}$$

where we are retaining the notation of expression (1.6), and where every component of $\underline{e} \in \mathcal{R}^n$ is one. It is straightforward to deduce the nonsingularity of this augmented system for the above choices of ϕ, using the properties of the symmetric matrix Φ that are stated in the previous paragraph.

This technique for including constant functions in $\mathcal{A}$ can be generalised to include all polynomials of degree m from $\mathcal{R}^d$ to $\mathcal{R}$, where m is a prescribed nonnegative integer. Specifically, letting p_j, $j=1,2,\ldots,\widehat{m}$, be a basis of this space of polynomials, expressions (1.8) and (1.9) are replaced by the function

$$s(\underline{x}) = \sum_{j=1}^{n} \lambda_j \, \phi(\|\underline{x}-\underline{x}_j\|) + \sum_{j=1}^{\widehat{m}} \alpha_j \, p_j(\underline{x}), \qquad \underline{x} \in \mathcal{R}^d, \tag{1.11}$$

and the constraints

$$\sum_{i=1}^{n} \lambda_i \, p_j(\underline{x}_i) = 0, \qquad j=1,2,\ldots,\widehat{m}, \tag{1.12}$$

respectively. Now the conditions (1.1) and (1.12) imply that the parameters λ_j, $j=1,2,\ldots,n$, and α_j, $j=1,2,\ldots,\widehat{m}$, have to satisfy an $(n+\widehat{m})\times(n+\widehat{m})$ system of linear equations, the main difference from the system (1.10) being that $\underline{e}$ and $\underline{e}^T$ are replaced by P and P^T, where P is the $n\times\widehat{m}$ matrix that has the elements $P_{ij} = p_j(\underline{x}_i)$, $i=1,2,\ldots,n$, $j=1,2,\ldots,\widehat{m}$. Singularity occurs if the rank of P is less than $\widehat{m}$, because then the last $\widehat{m}$ columns of the matrix of the system are linearly dependent. In this case, there exists a nonzero polynomial p of degree at most m that has the property $p(\underline{x}_i) = 0$, $i=1,2,\ldots,n$. Otherwise, it is known that the system is nonsingular for all the functions ϕ that have been mentioned.

The value $m=1$ is particularly useful in the technique that has just been described, because it allows some more choices of ϕ, including the thin plate spline case $\phi(r) = r^2 \log r$, $r \geq 0$, and the cubic case $\phi(r) = r^3$, $r \geq 0$. Then the nonsingularity of the $(n+\widehat{m})\times(n+\widehat{m})$ system of the previous paragraph is retained under the usual conditions, namely that the data points are all different and the rank of P is $\widehat{m}$. On the other hand, the thin plate spline method may fail if $m=0$. For example, if $n=d+1$, then, by placing the interpolation points at the vertices of a regular simplex in $\mathcal{R}^d$, all the distances $\|\underline{x}_i-\underline{x}_j\|$, $i\neq j$, can take the value one. Thus Φ becomes the $n\times n$ zero matrix, so the system (1.10) is singular. The zero matrix of this example is acceptable when $m=1$, however, because in this case the constraints (1.12) imply $\lambda_i=0$, $i=1,2,\ldots,n$. Therefore s is the unique linear (or constant) polynomial that interpolates the data $f(\underline{x}_i)$, $i=1,2,\ldots,d+1$.

It is instructive to consider the linear choice $\phi(r) = r$ and the cubic choice $\phi(r)=r^3$ when d (the number of variables) is one. Then equations (1.8) and (1.11) show that s is composed of polynomial pieces on the intervals between adjacent interpolation points. Thus radial basis function methods include piecewise linear and cubic spline interpolation to values of a function of one variable. Therefore, when $d > 1$, we can take the view that these choices of ϕ provide algorithms for multivariable interpolation that are extensions of piecewise polynomial methods in one dimension. On the other hand, when $d > 1$, the explicit use of piecewise polynomials introduces the problems of dividing $\mathcal{R}^d$ into regions that correspond to the different polynomial pieces, and ensuring that the function s has enough smoothness across the boundaries of the regions. Fortunately, these difficulties do not occur in radial basis function methods after ϕ has been chosen, so it is straightforward to allow d to be large.

When n is infinite, the interpolation points can be the vertices of an infinite square grid in $\mathcal{R}^d$. This setting is attractive theoretically, because the periodicity of the grid allows Fourier techniques to give some interesting results. It is usually necessary to rearrange the first sum of expression (1.11), as the given form may not be absolutely convergent. Further, the explicit polynomial part of expression (1.11) is deleted, because, for each ϕ that has been mentioned except the Gaussian, the usual polynomials are provided automatically by the radial basis function method. For example, in the case $\phi(r)=r$ and $d=1$, the piecewise linear form of $s(\underline{x})$, $\underline{x}\in\mathcal{R}$, can become a single linear polynomial when the interpolation points $\underline{x}_i$, $i=1,2,\ldots,n$, include divergent sequences in both the positive and negative directions on $\mathcal{R}$. A very important step in the analysis is the identification by Buhmann (1990) of the Fourier transforms of the Lagrange functions of interpolation on the grid $\mathcal{Z}^d$. Thus an integer $\check{m}$ is found, depending on ϕ and d, such that, if $f(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, is any polynomial of degree at most $\check{m}$, then interpolation on $\mathcal{Z}^d$ provides a function s that is equal to f identically. It follows from Taylor series expansions and the decay properties of the Lagrange functions that, if f has bounded $(\check{m}+1)$-th derivatives, then typically the error of the approximation $s\approx f$ has the magnitude

$$|\,f(\underline{x})-s(\underline{x})\,| = \mathcal{O}(h^{\check{m}+1}), \qquad \underline{x}\in\mathcal{R}^d, \tag{1.13}$$

where s is defined by interpolation to f on the infinite square grid of mesh size h. Some details have been omitted from this claim, however, such as relations between c and h when ϕ is the multiquadric $\phi(r)=(r^2+c^2)^{1/2}$. Further, another condition on f is sometimes necessary to avoid a factor of $|\log h|$ on the right hand side of equation (1.13).

The values of $\check{m}$ are brilliant, because $\check{m}$ increases with d for all the given choices of ϕ except the Gaussian. In particular, $\check{m}=d$ occurs when ϕ is a linear or multiquadric radial function, and the values $\check{m}=d+1$ and $\check{m}=d+2$ are achieved in the thin plate spline and cubic cases, respectively. Thus the bound (1.13) shows $\mathcal{O}(h^4)$ accuracy in three dimensions when $\phi(r)=r$, for example, which is difficult to accept, because of the first derivative discontinuities in s. The remark that these discontinuities do not prevent $\mathcal{O}(h^2)$ accuracy in one dimension throws some light on the situation. Moreover, the property (1.13) applies also to square grids that are finite in the linear, thin plate spline and cubic cases, subject to a restriction on $\underline{x}$ (Bejancu, 1999). Specifically, let the data points $\underline{x}_i$, $i=1,2,\ldots,n$, be all points of the unit cube $[0,1]^d$ whose components are integer multiples of h, where h is the reciprocal of a positive integer, and let $\mathcal{D}$ be any closed subset of the interior of the unit cube, for example the cube $[0.05,0.95]^d$. Then condition (1.13) is satisfied uniformly for $\underline{x}\in\mathcal{D}$ as $h\rightarrow 0$. On the other hand, the rate of convergence of s to f is less favourable near the boundaries of the finite grid, which is a well-known property of natural cubic spline interpolation in one dimension.

Some numerical experiments on the accuracy of linear and multiquadric radial basis function interpolation are reported in Section 2. They address the adequacy of the interpolation methods when d is large. Setting up and solving the system (1.10) for values of n up to about 10,000 is feasible for any dimension d, but n grows exponentially as d increases, if the interpolation points are all the vertices of a square grid in $\mathcal{R}^d$. In many applications, however, there are strong correlations between the components of relevant vectors $\underline{x}$. For example, $f(\underline{x})$ may be the life expectancy of a person who gives the answers x_i, $i=1,2,\ldots,d$, to d separate questions about his or her age, upbringing, health and wealth. Then, if d is large, the correlations usually cause the vectors $\underline{x}$ of the replies to occupy only a tiny part of $\mathcal{R}^d$. Therefore one may be able to reduce d by the elimination of redundant variables. On the other hand, if $s(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, is calculated by interpolation to the original data, then no information is wasted, so it is likely that s will be a useful approximation to f within the relevant part of $\mathcal{R}^d$. An advantage of the latter approach is that it avoids the decisions and solutions of equations that occur in elimination procedures. Therefore in Section 2 we investigate the size of the error $|f(\underline{x})-s(\underline{x})|$, $\underline{x}\in\mathcal{M}$, when the data points $\underline{x}_i$, $i=1,2,\ldots,n$, are all in $\mathcal{M}$, where $\mathcal{M}$ is a low dimensional nonlinear manifold in $\mathcal{R}^d$, and s is calculated by linear or multiquadric radial basis function interpolation. The accuracy that is found for moderate values of n is very encouraging.

In these experiments, the system of equations (1.10) is solved by an iterative algorithm that is described briefly in Section 3. The number of iterations depends strongly on the choice of a positive integer q, its purpose being that Lagrange functions of the given interpolation problem are approximated by Lagrange functions that employ only q of the data points $\underline{x}_i$, $i=1,2,\ldots,n$, as proposed by Beatson and Powell (1994). Therefore larger values of q usually reduce the number of iterations, but they increase the amount of work before the iterations are begun, and they make each iteration slightly more expensive. The value $q=30$ was found to be efficient during the development of the iterative algorithm, but then there were only two variables ($d=2$). Further, the calculations of Section 2 show a very welcome property, namely that this choice of q remains suitable when the interpolation points are on a two-dimensional manifold in $\mathcal{R}^d$ for large d. Details of the iterative method are given in Section 3, with some further investigations of the dependence of the number of iterations on q and the presence of manifolds. This dependence and some related questions on software for interpolation by radial basis functions are discussed in Section 4.

2. The accuracy of interpolation in high dimensions

Throughout this section, the interpolant s to the data $f(\underline{x}_i)$, $i=1,2,\ldots,n$, has the form

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

subject to the constraint $\sum_{j=1}^{n} \lambda_j = 0$. Therefore the conditions $c=0$ and $c>0$ on the parameter c provide the linear and multiquadric radial functions, respectively. We employ some numerical results to investigate the dependence of the error $|f(\underline{x})-s(\underline{x})|$, $\underline{x}\in\mathcal{D}$, on the dimension d, on the number and positions of the data points, and on c. The set $\mathcal{D}$, which is specified for each experiment, contains a finite number of suitable vectors $\underline{x}\in\mathcal{R}^d$. Values of the maximum error

$$\|f-s\|_\infty = \max\{|f(\underline{x})-s(\underline{x})| : \underline{x}\in\mathcal{D}\} \tag{2.2}$$

were calculated when f is a quadratic polynomial and when f is a Gaussian function, but the results of these two cases were found to be similar, except in the investigations of the last paragraph of this section. Therefore, until we reach that paragraph, we restrict attention to data functions of Gaussian type

$$f(\underline{x}) = \exp\left(-\theta\,\|\underline{x}-\underline{x}_0\|^2\right), \qquad \underline{x}\in\mathcal{R}^d, \tag{2.3}$$

where $\underline{x}_0$ is an appropriate centre of the interpolation points, and where θ is the positive number that is defined by the equation

$$\min\{f(\underline{x}_i) : i=1,2,\ldots,n\} = 0.01. \tag{2.4}$$

We are interested in the situation when the data points $\underline{x}_i$, $i=1,2,\ldots,n$, lie on a two or three dimensional manifold $\mathcal{M}\subset\mathcal{R}^d$, because we hope to find that, if $\mathcal{D}$ is a subset of $\mathcal{M}$, then the error (2.2) is small for moderate values of n, even if the dimension d is very large. The manifolds are constructed in the following way. We generate three vectors $\underline{a}^{(1)}$, $\underline{a}^{(2)}$ and $\underline{a}^{(3)}$ in $\mathcal{R}^d$, by setting their components to random numbers from the distribution that is uniform on $[0,2\pi]$. Then, for real variables ξ, η and ζ, we let $\underline{y}(\xi,\eta)$ and $\underline{z}(\xi,\eta,\zeta)$ be the vectors in $\mathcal{R}^d$ that have the components

$$\left.\begin{aligned} y_j &= \widetilde{\sin}(\xi+a_j^{(1)}) + \widetilde{\sin}(\eta+a_j^{(2)}) \\ z_j &= \widetilde{\sin}(\xi+a_j^{(1)}) + \widetilde{\sin}(\eta+a_j^{(2)}) + \widetilde{\sin}(\zeta+a_j^{(3)}) \end{aligned}\right\}, \quad j=1,2,\ldots,d, \tag{2.5}$$

where $\widetilde{\sin}$ is the approximation

$$\widetilde{\sin}(t) \;=\; (e^{\sin t}-1)\,/\,(e-1), \qquad t\in\mathcal{R}, \tag{2.6}$$

Edges	n	$d=5$	$d=10$	$d=20$	$d=40$
No	64	1.2×10^{-1}	6.7×10^{-2}	6.7×10^{-2}	5.7×10^{-2}
No	256	3.1×10^{-2}	1.1×10^{-2}	1.6×10^{-2}	1.1×10^{-2}
No	1024	4.1×10^{-3}	1.4×10^{-3}	2.2×10^{-3}	1.5×10^{-3}
No	4096	5.1×10^{-4}	1.8×10^{-4}	2.7×10^{-4}	1.8×10^{-4}
Yes	64	3.1×10^{-2}	4.0×10^{-2}	3.3×10^{-2}	5.1×10^{-2}
Yes	256	5.8×10^{-3}	4.5×10^{-3}	3.8×10^{-3}	5.5×10^{-3}
Yes	1024	1.2×10^{-3}	5.7×10^{-4}	6.6×10^{-4}	6.3×10^{-4}
Yes	4096	3.4×10^{-4}	2.7×10^{-4}	3.2×10^{-4}	3.2×10^{-4}

Table 1: $\|f-s\|_\infty$ for regular data on a two dimensional manifold

to the sine function. Further, we let the two and three dimensional manifolds $\mathcal{M}\subset\mathcal{R}^d$ be the sets

$$\{\underline{y}(\xi,\eta) : 0 \le \xi,\eta \le 2\pi\} \quad \text{and} \quad \{\underline{z}(\xi,\eta,\zeta) : 0 \le \xi,\eta,\zeta \le 2\pi\}, \tag{2.7}$$

respectively. Thus the elements of $\mathcal{M}$ span $\mathcal{R}^d$, but, if "$\widetilde{\sin}$" were replaced by "sin" in the definition (2.5), then $\underline{y}(\xi,\eta)$ and $\underline{z}(\xi,\eta,\zeta)$ would be confined to subspaces of $\mathcal{R}^d$ of dimension at most 4 and 6, respectively. It follows from the periodicity of the sine function that the given manifolds have no edges and are bounded. Therefore they are very different from linear manifolds.

These choices of $\mathcal{M}$ have the unusual advantage of allowing the points $\underline{x}_i$, $i=1,2,\ldots,n$, to be placed on a finite square grid of the variables of the manifold without any edge effects. This is done in the first of our experiments, $\mathcal{M}$ being two dimensional, and each n being the square of an integer, m say. Specifically, we choose the interpolation points

$$\{\,\underline{x}_i : i=1,2,\ldots,n\,\} \;=\; \{\,\underline{y}(jh_1,kh_1) : 1 \le j,k \le m\,\}, \tag{2.8}$$

where h_1 is the grid size $2\pi/m$. Further, the parameter c of the interpolant (2.1) is set to zero until we reach Table 3, the elements of $\mathcal{D}$ in expression (2.2) are the centres of the grid squares, which form the set

$$\mathcal{D} \;=\; \{\,\underline{y}(jh_1-\tfrac{1}{2}h_1,\, kh_1-\tfrac{1}{2}h_1) : 1 \le j,k \le m\,\}, \tag{2.9}$$

and $\underline{x}_0$ is given the value $\underline{y}(\pi,\pi)$ in the function (2.3). The resultant errors $\|f-s\|_\infty$ for several choices of d and n are shown in the top half of Table 1. We also investigate interpolation on a square grid that has edges, the results being given in the bottom half of the table. Now the grid size is reduced to $h_2=\pi/(m-1)$, and we pick the points

$$\{\,\underline{x}_i : i=1,2,\ldots,n\,\} \;=\; \{\,\underline{y}(jh_2,kh_2) : 0 \le j,k \le m-1\,\}. \tag{2.10}$$

Edges	n	$d=5$	$d=10$	$d=20$	$d=40$
No	64	9.6×10^{-2}	8.1×10^{-2}	8.0×10^{-2}	5.6×10^{-2}
No	512	1.8×10^{-2}	2.5×10^{-2}	4.1×10^{-2}	4.5×10^{-2}
No	4096	2.3×10^{-3}	2.0×10^{-3}	3.8×10^{-3}	5.3×10^{-3}
Yes	64	4.8×10^{-2}	7.6×10^{-2}	2.7×10^{-1}	2.6×10^{-1}
Yes	512	6.5×10^{-3}	6.0×10^{-3}	1.6×10^{-2}	1.5×10^{-2}
Yes	4096	1.8×10^{-3}	1.6×10^{-3}	1.3×10^{-3}	1.1×10^{-3}

Table 2: $\|f-s\|_\infty$ for regular data on a three dimensional manifold

Further, $\mathcal{D}$ is the set $\{\, \underline{y}(jh_2-\frac{1}{2}h_2, kh_2-\frac{1}{2}h_2) : 1\le j,k\le m-1 \,\}$ of centres of the grid squares, and $\underline{x}_0$ is the vector $\underline{y}(\frac{1}{2}\pi, \frac{1}{2}\pi)$. We see in the table that, for both of these experiments, the accuracy of the approximation $s\approx f$ remains about the same when d is increased, but increases in n cause substantial reductions in the error. These findings are very welcome. There is also excellent agreement with a property of interpolation on the infinite grid $h\mathcal{Z}^d$ that is mentioned in Section 1. Specifically, the bound (1.13) with $\check{m}=d$ suggests that $\|f-s\|_\infty$ should decrease by a factor of eight when the grid size is halved if the grid is without edges. The No Edge entries of Table 1 when $n=1024$ and $n=4096$ provide strong support for this conjecture. On the other hand, the numbers in the last two rows suggest the magnitude $\|f-s\|_\infty=\mathcal{O}(h_2)$ when edges are present, h_2 being the grid size.

Table 2 gives values of $\|f-s\|_\infty$ for three dimensional manifolds that are analogous to the results of Table 1. Now n is the cube of an integer m, so we retain the mesh sizes $h_1=2\pi/m$ and $h_2=\pi/(m-1)$ for the regular grids without and with edges, respectively. Therefore the sets of interpolation points in these cases are $\underline{z}(jh_1, kh_1, \ell h_1)$, $1\le j,k,\ell\le m$ and $\underline{z}(jh_2, kh_2, \ell h_2)$, $0\le j,k,\ell\le m-1$. Similarly, the choices of $\underline{x}_0$ are $\underline{z}(\pi,\pi,\pi)$ and $\underline{z}(\frac{1}{2}\pi, \frac{1}{2}\pi, \frac{1}{2}\pi)$, respectively, and the sets $\mathcal{D}$ contain the vectors $\underline{z}(\xi,\eta,\zeta)$, where each (ξ,η,ζ) is a centre of a grid cube of the manifold. The main feature of Table 2 is confirmation that a few thousand data points can provide good accuracy when interpolating smooth functions of many variables on a three dimensional manifold. It seems, however, that there are not enough data points in these experiments to show the asymptotic behaviour of $\|f-s\|_\infty$ as the mesh size tends to zero. The calculations with edges are easier in one way, which is that interpolation and measurement of the error is confined to only one eighth of the full manifold $\mathcal{M}$. Therefore many of the entries in the lower half of Table 2 are less than the corresponding entries in the upper half, but this situation should change for larger values of n.

In some other experiments, the positions of the interpolation points $\underline{x}_i$, $i=1,2,\ldots,n$, are generated randomly on $\mathcal{M}$, instead of being at vertices of a regular grid of the variables of the manifold. For example, when $\mathcal{M}$ is two dimensional, we set $\underline{x}_i = \underline{y}(\xi,\eta)$, where ξ and η are random numbers from $[0,2\pi]$ for each integer i in $[1,n]$, but we take the choices of $\mathcal{D}$ and $\underline{x}_0 = \underline{y}(\pi,\pi)$ from the calculations of the first half of Table 1, the number $m = n^{1/2}$ being an integer as before. The new values of $\|f-s\|_\infty$ for $n=64$ are 1.7×10^{-1}, 1.4×10^{-1}, 1.6×10^{-1} and 5.8×10^{-2} in the cases $d=5$, 10, 20 and 40, respectively, which are not much larger than the entries for regular data in the first row of Table 1. When $n=4096$, however, the new values of $\|f-s\|_\infty$ are between 9.9×10^{-4} and 4.7×10^{-3}, but the greatest entry in the fourth row of Table 1 is only 5.1×10^{-4}. We also employ random interpolation points in some experiments that are comparable to the lower half of Table 1, each $\underline{x}_i$ having the form $\underline{y}(\xi,\eta)$, where the variables ξ and η are sampled from $[0,\pi]$. Further, we retain $\underline{x}_0 = \underline{y}(\frac{1}{2}\pi, \frac{1}{2}\pi)$, but, in order to avoid large contributions to $\|f-s\|_\infty$ from edge effects, we let $\mathcal{D}$ contain just 1000 vectors $\underline{y}(\xi,\eta)$, these variables ξ and η being sampled randomly from the interval $[0.05\pi, 0.95\pi]$. The resulting values of $\|f-s\|_\infty$ for $d=5$, 10, 20 and 40 are between 4.9×10^{-2} and 8.8×10^{-2} when $n=64$, and between 2.9×10^{-4} and 3.9×10^{-4} when $n=4096$. Therefore the increase in n achieves reductions in the errors of interpolation that are like the reductions that are shown in the second half of Table 1. Versions of the calculations of this paragraph for three dimensional manifolds were also tried. The values of $\|f-s\|_\infty$ for random interpolation points on the full manifold are between 8.8×10^{-2} and 2.2×10^{-1} when $n=64$, and between 4.9×10^{-3} and 1.6×10^{-2} when $n=4096$. They should be compared with the entries in the first and third rows of Table 2. Furthermore, the results when the interpolation points are chosen randomly from the set $\{\underline{z}(\xi,\eta,\zeta) : 0 \le \xi,\eta,\zeta \le \pi\}$, which is one eighth of $\mathcal{M}$, are between 5.4×10^{-2} and 2.3×10^{-1} when $n=64$, and between 1.5×10^{-3} and 2.2×10^{-3} when $n=4096$, so they are similar to the corresponding values in the lower half of Table 2. The given figures are only an indication of accuracy, however, as they depend strongly on the random numbers that occur.

It is often advantageous to let the parameter c of the interpolant (2.1) be positive, not only because derivatives of s become continuous, but also because, when f is smooth, the error $\|f-s\|_\infty$ decreases usually if c is increased from zero. Therefore Table 3 shows extensions of some of the Table 1 results to positive values of c. Expression (2.1) suggests that typical distances between nearest neighbours in the set $\{\underline{x}_i : i=1,2,\ldots,n\}$ are a suitable magnitude for c, these distances being approximately $h_1 d^{1/2}$ and $h_2 d^{1/2}$ in the cases (2.8) and (2.10), respectively. Further, in order to try several choices of c, these distances are scaled by the factor σ that is included in Table 3. The "No" headings of the columns of the table denote no edge effects, so their calculations are identical to the corresponding calculations of the first half of Table 1, except that c is set to $\sigma h_1 d^{1/2}$. Similarly, the calculations of the "Yes" columns of Table 3 are identical to the calculations of the "Yes" rows of Table 1, except for the setting $c = \sigma h_2 d^{1/2}$.

The values of $\|f-s\|_\infty$ in Table 3 show clearly that, for fixed data, the error

n	σ	$d\!=\!5$ (No)	$d\!=\!40$ (No)	$d\!=\!5$ (Yes)	$d\!=\!40$ (Yes)
64	0.0	1.2×10^{-1}	5.7×10^{-2}	3.1×10^{-2}	5.1×10^{-2}
64	0.5	1.8×10^{-2}	8.1×10^{-3}	4.5×10^{-3}	4.4×10^{-3}
64	1.0	1.8×10^{-2}	1.0×10^{-2}	3.4×10^{-3}	1.6×10^{-3}
64	2.0	3.8×10^{-2}	1.3×10^{-2}	3.0×10^{-3}	1.7×10^{-3}
4096	0.0	5.1×10^{-4}	1.8×10^{-4}	3.4×10^{-4}	3.2×10^{-4}
4096	0.5	2.1×10^{-5}	2.4×10^{-6}	3.4×10^{-4}	2.9×10^{-4}
4096	1.0	3.3×10^{-6}	1.7×10^{-8}	1.5×10^{-4}	1.2×10^{-4}
4096	2.0	3.0×10^{-7}	3.8×10^{-12}	2.6×10^{-5}	1.9×10^{-5}

Table 3: $\|f-s\|_\infty$ on a two dimensional manifold for $c\geq 0$

of the approximation $s\approx f$ can often be reduced greatly by letting c be positive instead of zero. The gains are stunning in the cases with no edge effects and 4096 data points, which is related to the bound (1.13) on the error of interpolation on the infinite grid $h\mathcal{Z}^d$. Indeed, stronger bounds apply for the multiquadric radial function $\phi(r)=(r^2+c^2)^{1/2}$, $r\geq 0$, if c tends to zero more slowly than h. It follows that, if h is sufficiently small and fixed, then s can enjoy the higher order accuracy by increasing c from a value of magnitude h. The large gains in Table 3 are an example of this remark. On the other hand, an unfavourable property of greater values of c is that the value of $s(\underline{x})$ for any $\underline{x}$ tends to be influenced more by data at points that are remote from $\underline{x}$. Therefore we find also in Table 3 that some of the increases in σ cause a deterioration in the value of $\|f-s\|_\infty$.

It is instructive to compare the experiments so far with cases when the data points $\underline{x}_i$, $i=1,2,\ldots,n$, do not lie on a low dimensional manifold in $\mathcal{R}^d$. Therefore we let these points be sampled randomly from the distribution that is uniform throughout the unit ball $\{\underline{x} : \|\underline{x}\| \leq 1\} \subset \mathcal{R}^d$, and we let $\underline{x}_0$ be the origin of $\mathcal{R}^d$. Unfortunately, however, if we retain the function (2.3) and if $d=40$, then the Gaussian peak of f at $\underline{x}_0$ is too narrow to be shown adequately by the data unless n is huge. Indeed, in a test with $n=10^6$, the conditions $f(\underline{x}_i)\geq 0.45$ and $f(\underline{x}_i)\geq 0.5$ were satisfied for only 32 and 0 values of i, respectively. Therefore, for the remainder of this section, we switch to the quadratic data function

$$f(\underline{x}) \;=\; \tfrac{1}{4}\,\|\underline{x}-\underline{e}_1\|^2, \qquad \underline{x}\in\mathcal{R}^d, \tag{2.11}$$

where $\underline{e}_1$ is the first coordinate vector.

Our choice of the set $\mathcal{D}$ for expression (2.2) in the new experiments is 1000 random vectors from the distribution that is uniform in the ball $\{\underline{x} : \|\underline{x}\|\leq 0.9\}$ of $\mathcal{R}^d$. We employ randomness, because a grid of points would be too expensive, and the purpose of the smaller ball is to avoid domination of $\|f-s\|_\infty$ by edge effects.

n	σ	$d=5$	$d=10$	$d=20$	$d=40$
64	0.0	5.9×10^{-2}	1.2×10^{-1}	1.3×10^{-1}	1.9×10^{-1}
64	0.5	4.2×10^{-2}	1.1×10^{-1}	1.1×10^{-1}	1.7×10^{-1}
64	1.0	3.2×10^{-2}	9.5×10^{-2}	1.1×10^{-1}	1.6×10^{-1}
64	2.0	1.9×10^{-2}	6.5×10^{-2}	1.1×10^{-1}	1.4×10^{-1}
4096	0.0	5.6×10^{-3}	1.8×10^{-2}	2.0×10^{-2}	3.6×10^{-2}
4096	0.5	4.2×10^{-3}	1.4×10^{-2}	2.1×10^{-2}	2.2×10^{-2}
4096	1.0	2.8×10^{-3}	8.4×10^{-3}	1.6×10^{-2}	1.3×10^{-2}
4096	2.0	1.3×10^{-3}	1.8×10^{-3}	7.2×10^{-3}	5.0×10^{-3}

Table 4: $\|f-s\|_\infty$ for random interpolation points in the unit ball

The parameter c of the function (2.1) is set to the product $c=\sigma(1/n)^{1/d}$ for the range of values of σ that has been used already, the factor $(1/n)^{1/d}$ being the magnitude of nearest neighbour distances between data points. A selection of the values of $\|f-s\|_\infty$ that were found is given in Table 4. The set $\mathcal{D}$ was not altered during the calculations of each column of the table, and the set $\{\underline{x}_i : i=1,2,\ldots,n\}$ was not changed when σ was increased for each d and n. The table suggests that, when the points $\underline{x}_i$, $i=1,2,\ldots,n$, are in general positions in $\mathcal{R}^d$, then the radial basis function interpolation method may be useful sometimes, even if f is a function of 40 variables. We see also that some gains in accuracy can still be achieved by increasing c, the gains being greater for the larger value of n, but the success of this technique for large d requires f to be very smooth.

3. The solution of the interpolation equations

In the experiments of Section 2, the system of equations (1.10) is solved by the iterative algorithm of Faul, Goodsell and Powell (2002). It is described briefly in this section. For $k=1,2,3,\ldots$, the k-th iteration of the algorithm begins with a function $s^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, of the form (1.8), which is the current estimate of the required interpolant s. The vector $\underline{r}^{(k)}\in\mathcal{R}^n$, whose components are the residuals

$$r_i^{(k)} = f(\underline{x}_i) - s^{(k)}(\underline{x}_i), \qquad i=1,2,\ldots,n, \tag{3.1}$$

is also available. By combining these residuals with the coefficients of the approximations to Lagrange functions that are mentioned at the end of Section 1, a reasonable change to $s^{(k)}$, namely $t^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, is generated. It is used in a procedure that is analogous to the conjugate gradient method. Specifically, a direction function $d^{(k)}$ is given by the formula

$$d^{(k)}(\underline{x}) = \begin{cases} t^{(1)}(\underline{x}), & k=1, \\ t^{(k)}(\underline{x}) - \beta_k\, d^{(k-1)}(\underline{x}), & k\geq 2, \end{cases} \qquad \underline{x}\in\mathcal{R}^d, \tag{3.2}$$

where β_k is a parameter that makes $d^{(k)}$ conjugate to $d^{(k-1)}$, the meaning of conjugacy being given later. By searching from $s^{(k)}$ along the direction $d^{(k)}$, the new estimate

$$s^{(k+1)}(\underline{x}) = s^{(k)}(\underline{x}) + \gamma_k\, d^{(k)}(\underline{x}), \qquad \underline{x}\in\mathcal{R}^d, \tag{3.3}$$

of s is found, where the step-length γ_k is also defined later. Then the residuals $r_i^{(k+1)} = f(\underline{x}_i) - s^{(k+1)}(\underline{x}_i)$, $i=1,2,\ldots,n$, are calculated, and termination occurs if they can be made sufficiently small by adding a constant to $s^{(k+1)}$. Otherwise, k is increased by one for the next iteration. The choice $s^{(1)}(\underline{x})=0$, $\underline{x}\in\mathcal{R}^d$, is made initially, so the components of the first residual vector $\underline{r}^{(1)}$ are the data $f(\underline{x}_i)$, $i=1,2,\ldots,n$. The convergence properties of the algorithm are addressed by Faul and Powell (2000).

Our notation for the coefficients of the radial basis function terms of $s^{(k)}$, $t^{(k)}$ and $d^{(k)}$ is shown in the expression

$$\left.\begin{array}{rcl} s^{(k)}(\underline{x}) &=& \sum_{j=1}^n \sigma_j^{(k)} \phi(\|\underline{x}-\underline{x}_j\|) + \text{constant} \\ t^{(k)}(\underline{x}) &=& \sum_{j=1}^n \tau_j^{(k)} \phi(\|\underline{x}-\underline{x}_j\|) + \text{constant} \\ d^{(k)}(\underline{x}) &=& \sum_{j=1}^n \delta_j^{(k)} \phi(\|\underline{x}-\underline{x}_j\|) + \text{constant} \end{array}\right\}, \qquad \underline{x}\in\mathcal{R}^d. \tag{3.4}$$

The algorithm works with $\sigma_j^{(k)}$, $\tau_j^{(k)}$ and $\delta_j^{(k)}$, $j=1,2,\ldots,n$, explicitly. They are generated in ways that satisfy the conditions

$$\sum_{j=1}^n \sigma_j^{(k)} = \sum_{j=1}^n \tau_j^{(k)} = \sum_{j=1}^n \delta_j^{(k)} = 0, \qquad k=1,2,3,\ldots, \tag{3.5}$$

because of the constraint (1.9) on the coefficients of s. In other words, for each of the functions (3.4), the radial basis function coefficients are the components of a vector in an $n-1$ dimensional subspace $\mathcal{S}$ of $\mathcal{R}^n$, where $\underline{v}\in\mathcal{R}^n$ is in $\mathcal{S}$ if and only if the sum of its components is zero. Now, because we are considering the linear and multiquadric choices of ϕ, we recall from the paragraph that includes equation (1.7) that the quadratic form $\underline{v}^T\Phi\underline{v}$, $\underline{v}\in\mathcal{S}$, is negative if $\underline{v}$ is nonzero. This remark provides a semi-norm for the n dimensional linear space $\mathcal{A}$ of functions of the form (1.8), subject to the constraints (1.9), assuming that ϕ and $\underline{x}_j$, $j=1,2,\ldots,n$, are prescribed. Specifically, we let s have the semi-norm $\|s\|_\phi = [-\underline{\lambda}^T\Phi\underline{\lambda}]^{1/2}$, so $s^{(k)}$, for example, has the semi-norm

$$\|s^{(k)}\|_\phi = [-\underline{\sigma}^{(k)T}\Phi\underline{\sigma}^{(k)}]^{1/2} = \Big[-\sum_{i=1}^n\sum_{j=1}^n \sigma_i^{(k)}\sigma_j^{(k)}\phi(\|\underline{x}_i-\underline{x}_j\|)\Big]^{1/2}. \tag{3.6}$$

There is also a scalar product that corresponds to the semi-norm. Indeed, if $\underline{u}$ and $\underline{v}$ are the vectors of radial basis function coefficients of any two functions in $\mathcal{A}$, then these functions have the scalar product $-\underline{u}^T\Phi\underline{v}$. Therefore the last line of equation (3.4) gives the value

$$\langle d^{(k-1)}, d^{(k)}\rangle_\phi = -\underline{\delta}^{(k-1)T}\Phi\underline{\delta}^{(k)} = -\sum_{i=1}^{n}\sum_{j=1}^{n} \delta_i^{(k-1)}\delta_j^{(k)}\phi(\|\underline{x}_i - \underline{x}_j\|), \tag{3.7}$$

where this left hand side shows our notation for the scalar product between $d^{(k-1)}$ and $d^{(k)}$. We say that $d^{(k)}$ is conjugate to $d^{(k-1)}$ if the scalar product (3.7) is zero.

We also use equation (3.4) to remove the sums over j from expressions (3.6) and (3.7), noting that the constraints (3.5) eliminate the contributions from the constant terms. Thus we deduce the formulae

$$\left.\begin{array}{rcl} \|s^{(k)}\|_\phi & = & \left[-\sum_{i=1}^n \sigma_i^{(k)} s^{(k)}(\underline{x}_i)\right]^{1/2} \\ \langle d^{(k-1)}, d^{(k)}\rangle_\phi & = & -\sum_{i=1}^n \delta_i^{(k-1)} d^{(k)}(\underline{x}_i) \end{array}\right\}, \tag{3.8}$$

which are very helpful to the algorithm as they avoid some matrix times vector multiplications. More importantly, this removal of Φ allows the scalar products between the required interpolant s and the functions (3.4) to be calculated, because the values $s(\underline{x}_i) = f(\underline{x}_i)$, $i=1,2,\ldots,n$, are data. In particular, the algorithm employs the identity

$$\langle d^{(k)}, s\rangle_\phi = -\sum_{i=1}^n \delta_i^{(k)} s(\underline{x}_i) = -\sum_{i=1}^n \delta_i^{(k)} f(\underline{x}_i). \tag{3.9}$$

For $k \geq 2$, the scalar product (3.7) is zero as required, if the parameter β_k of formula (3.2) is the quotient

$$\beta_k \;=\; \frac{\langle t^{(k)}, d^{(k-1)}\rangle_\phi}{\langle d^{(k-1)}, d^{(k-1)}\rangle_\phi} \;=\; \frac{\sum_{i=1}^n \tau_i^{(k)} d^{(k-1)}(\underline{x}_i)}{\sum_{i=1}^n \delta_i^{(k-1)} d^{(k-1)}(\underline{x}_i)}. \tag{3.10}$$

Moreover, the parameter γ_k of the function (3.3) is chosen to minimise the semi-norm $\|s - s^{(k+1)}\|_\phi$, where s is still the required interpolant. Therefore it has the value

$$\gamma_k \;=\; \frac{\langle d^{(k)}, s - s^{(k)}\rangle_\phi}{\langle d^{(k)}, d^{(k)}\rangle_\phi} \;=\; \frac{\sum_{i=1}^n \delta_i^{(k)} r_i^{(k)}}{\sum_{i=1}^n \delta_i^{(k)} d^{(k)}(\underline{x}_i)}, \tag{3.11}$$

the last numerator being derived from equations (3.9) and (3.1). The algorithm calculates expressions (3.10) and (3.11) explicitly, the function values $d^{(k)}(\underline{x}_i)$, $i = 1,2,\ldots,n$, being set to the components of the product $\Phi\underline{\delta}^{(k)}$, because the constant term of $d^{(k)}(\underline{x})$, $\underline{x} \in \mathcal{R}^d$, in equation (3.4) does not matter. Only one vector is multiplied by Φ on each iteration, which is important because this is the most expensive part of the current implementation of the algorithm when n

is large. Equations (3.3) and (3.1) show that the coefficients of $s^{(k+1)}$ and the components of $\underline{r}^{(k+1)}$ are given by the formulae

$$\left.\begin{array}{rcll} \sigma_j^{(k+1)} & = & \sigma_j^{(k)} + \gamma_k\, \delta_j^{(k)}, & j=1,2,\ldots,n \\ r_i^{(k+1)} & = & r_i^{(k)} - \gamma_k\, d^{(k)}(\underline{x}_i), & i=1,2,\ldots,n \end{array}\right\}, \tag{3.12}$$

except that a constant may be added to the new residuals, as mentioned in the first paragraph of this section. Most of these details belong to the conjugate gradient method, and next we turn to the choice of $t^{(k)}(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, which can be regarded as the gradient vector of the conjugate gradient method, after applying a preconditioner that improves the convergence of the iterative algorithm.

Equations (3.4) and (3.5) show that $t^{(k)}$ is in the linear space $\mathcal{A}$ that has been mentioned. Further, we let the constant term of $t^{(k)}$ be zero, as it is irrelevant to the parts of the algorithm that have been described already. Specifically, we express $t^{(k)}$ as the sum

$$t^{(k)}(\underline{x}) = \sum_{\ell=1}^{n-1} \mu_\ell^{(k)} z_\ell(\underline{x}), \qquad \underline{x}\in\mathcal{R}^d, \tag{3.13}$$

where the functions z_ℓ, $\ell=1,2,\ldots,n-1$, are linearly independent elements of $\mathcal{A}$ with zero constant terms. Therefore z_ℓ has the form

$$z_\ell(\underline{x}) = \sum_{j=1}^{n} \zeta_{\ell j}\, \phi(\|\underline{x}-\underline{x}_j\|), \qquad \underline{x}\in\mathcal{R}^d, \tag{3.14}$$

the sum $\sum_{j=1}^n \zeta_{\ell j}$ being zero. Particular choices of z_ℓ, $\ell=1,2,\ldots,n-1$, will be made later.

For each ℓ, the parameter $\mu_\ell^{(k)}$ of the sum (3.13) is defined by regarding z_ℓ as a search direction from the estimate $s^{(k)}$ of the interpolant s. Therefore we pick $\mu_\ell^{(k)}$ so that it minimises the semi-norm $\|s-s^{(k)}-\mu_\ell^{(k)} z_\ell\|_\phi$. It follows from the derivation of equation (3.11) that this parameter has the value

$$\mu_\ell^{(k)} \;=\; \frac{\langle z_\ell, s-s^{(k)}\rangle_\phi}{\langle z_\ell, z_\ell\rangle_\phi} \;=\; \frac{\sum_{i=1}^n \zeta_{\ell i}\, r_i^{(k)}}{\sum_{i=1}^n \zeta_{\ell i}\, z_\ell(\underline{x}_i)}, \qquad \ell=1,2,\ldots,n-1. \tag{3.15}$$

This construction and the independence of z_ℓ, $\ell=1,2,\ldots,n-1$, imply that, if $\|s-s^{(k)}\|_\phi$ is nonzero, then $\|s-s^{(k)}-\gamma\, t^{(k)}\|_\phi$, $\gamma\in\mathcal{R}$, decreases initially as γ is increased from zero. Further, the choice (3.2) of $d^{(k)}$ ensures that $\|s-s^{(k)}-\gamma\, d^{(k)}\|_\phi$, $\gamma\in\mathcal{R}$, has this property too, which is crucial to the convergence of the algorithm.

Termination would occur at the end of the first iteration if the functions z_ℓ, $\ell=1,2,\ldots,n-1$, satisfied the orthogonality conditions

$$\langle z_j, z_\ell\rangle_\phi = 0, \qquad 1\le j<\ell\le n-1. \tag{3.16}$$

Indeed, in this case expression (3.13) for $t^{(k)}$ and the values (3.15) would provide the scalar products

$$\langle z_j, s-s^{(k)}-t^{(k)}\rangle_\phi = \langle z_j, s-s^{(k)}-\mu_j^{(k)} z_j\rangle_\phi = 0, \qquad j=1,2,\ldots,n-1. \tag{3.17}$$

Hence, because $s-s^{(k)}-t^{(k)}$ is a linear combination of z_j, $j=1,2,\ldots,n-1$, it follows that $\|s-s^{(k)}-t^{(k)}\|_\phi$ would be zero. Thus the choice $d^{(k)}=t^{(k)}$ on the first iteration, and the calculation of γ_k to minimise $\|s-s^{(k)}-\gamma_k\, d^{(k)}\|_\phi$, would give the value

$$\|s-s^{(2)}\|_\phi = \|s-s^{(1)}-\gamma_1\, d^{(1)}\|_\phi = \|s-s^{(1)}-\gamma_1\, t^{(1)}\|_\phi = 0. \tag{3.18}$$

In other words, the difference $s(\underline{x})-s^{(2)}(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, would be a constant function. Then, by adding this constant to $s^{(2)}$, the algorithm would achieve the values $s^{(2)}(\underline{x}_i)=s(\underline{x}_i)=f(\underline{x}_i)$, $i=1,2,\ldots,n$, so $s^{(2)}$ would be the required interpolant.

The selection of the functions z_ℓ, $\ell=1,2,\ldots,n-1$, is guided by the remark that some Lagrange functions enjoy the orthogonality conditions (3.16). The first of these Lagrange functions, $\check{z}_1$ say, is the unique element of $\mathcal{A}$ that satisfies the equations $\check{z}_1(\underline{x}_i)=\delta_{i1}$, $i=1,2,\ldots,n$, where the right hand side is the Kronecker delta, and usually the constant term of $\check{z}_1$ is nonzero. Further, for $\ell=1,2,\ldots,n$–1, we let $\check{z}_\ell$ be the unique function of the form

$$\check{z}_\ell(\underline{x}) = \sum_{j=\ell}^{n} \check{\zeta}_{\ell j}\, \phi(\|\underline{x}-\underline{x}_j\|) + \text{constant}, \qquad \underline{x}\in\mathcal{R}^d, \tag{3.19}$$

that has the properties

$$\sum_{j=\ell}^{n} \check{\zeta}_{\ell j} = 0 \qquad \text{and} \qquad \check{z}_\ell(\underline{x}_i) = \delta_{i\ell}, \qquad i=\ell,\ell+1,\ldots,n. \tag{3.20}$$

Thus we find the scalar product

$$\langle \check{z}_m, \check{z}_\ell\rangle_\phi = -\sum_{i=m}^{n}\sum_{j=\ell}^{n} \check{\zeta}_{mi}\,\check{\zeta}_{\ell j}\,\phi(\|\underline{x}_i-\underline{x}_j\|) = -\sum_{i=m}^{n} \check{\zeta}_{mi}\,\check{z}_\ell(\underline{x}_i), \tag{3.21}$$

which is zero for $m>\ell$, because equation (3.20) shows that the numbers $\check{z}_\ell(\underline{x}_i)$, $i=m,m+1,\ldots,n$, are zero. Therefore this construction provides the conditions

$$\langle \check{z}_\ell, \check{z}_m\rangle_\phi = 0, \qquad 1\le \ell < m \le n-1, \tag{3.22}$$

as required. Of course the coefficients of $\check{z}_1$ are not computed when n is large, because this task would require almost as much work as the solution of the given interpolation equations.

Instead, we adopt the suggestion of Beatson and Powell (1994) that it may be adequate to employ approximations to the functions $\check{z}_\ell$, $\ell=1,2,\ldots,n-q$, where each approximation depends on the positions of only q of the data points, as stated in the last paragraph of Section 1. Specifically, q is prescribed, and the function z_ℓ is chosen in the following way for each integer ℓ in $[1,n-1]$. A set $\mathcal{L}_\ell \subset \{\ell,\ell+1,\ldots,n\}$ is formed, which is just $\{\ell,\ell+1,\ldots,n\}$ in the case $\ell > n-q$. Otherwise $\mathcal{L}_\ell$ contains q different integers from the interval $[\ell,n]$, which are selected so that the distances $\|\underline{x}_j - \underline{x}_\ell\|$, $j \in \mathcal{L}_\ell$, are as small as possible. Then the coefficients $\zeta_{\ell j}$, $j \in \mathcal{L}$, are given the values that make the function

$$\widehat{z}_\ell(\underline{x}) = \sum_{j\in\mathcal{L}_\ell} \zeta_{\ell j}\, \phi(\|\underline{x}-\underline{x}_j\|) + \text{constant}, \qquad \underline{x} \in \mathcal{R}^d, \tag{3.23}$$

satisfy the conditions

$$\sum_{j\in\mathcal{L}_\ell} \zeta_{\ell j} = 0 \qquad \text{and} \qquad \widehat{z}_\ell(\underline{x}_i) = \delta_{i\ell}, \qquad i \in \mathcal{L}_\ell. \tag{3.24}$$

The other coefficients of expression (3.14), namely $\zeta_{\ell j}$, $j \in \{1,2,\ldots,n\} \backslash \mathcal{L}_\ell$, are set to zero, which completes the definition of z_ℓ. In other words, z_ℓ is the function (3.23) without the constant term. An advantage of this method is that equation (3.15) reduces to the formula

$$\mu_\ell^{(k)} = \sum_{i\in\mathcal{L}_\ell} \zeta_{\ell i}\, r_i^{(k)} \Big/ \zeta_{\ell\ell}, \qquad \ell = 1,2,\ldots,n-1. \tag{3.25}$$

Further, the calculation of the coefficients $\tau_j^{(k)}$, $j=1,2,\ldots,n$, of the function (3.13) takes only $\mathcal{O}(nq)$ computer operations on each iteration.

However, the work of generating the numbers $\zeta_{\ell j}$, $j \in \mathcal{L}_\ell$, $\ell = 1,2,\ldots,n-1$, initially is of magnitude nq^3. Moreover, in the present version of the algorithm, the calculation of the product $\Phi\underline{\delta}^{(k)}$ on each iteration, mentioned soon after equation (3.11), requires $\mathcal{O}(n^2)$ operations. Therefore the influence of q on the number of iterations is very important to efficiency. It is investigated numerically in the remainder of this section when the number of variables is large, particular attention being given to the case when the data points $\underline{x}_i$, $i=1,2,\ldots,n$, lie on a low dimensional manifold in $\mathcal{R}^d$. Another question is whether efficiency can be impaired by the influence of the ordering of the data points on the method of the previous paragraph. Numerical testing has shown that it is disadvantageous if the sequence $\underline{x}_i$, $i=1,2,\ldots,n$, tends to track gradually through the region of $\mathcal{R}^d$ that contains the data. Therefore the data points are rearranged randomly at the beginning of the calculation. Many properties of the algorithm, including relations to the conjugate gradient method, are studied by Faul (2001).

The experiments at the end of Section 2, where the points $\underline{x}_i$, $i=1,2,\ldots,n$, are not on a manifold, provide some interesting tests of the iterative algorithm, because it may be impossible to construct good approximations to Lagrange functions in $\mathcal{R}^d$, using a small number of data points when d is large. Therefore the line search of the method, in other words the choice of γ_k in formula (3.3), has to compensate for the inadequacies of the approximations. Thus the algorithm

n	q	$d=2$	$d=5$	$d=10$	$d=20$	$d=40$
256	10	13	29	29	33	31
256	30	6	17	28	31	28
256	50	5	11	21	29	28
1024	10	16	53	61	54	54
1024	30	8	30	54	53	44
1024	50	6	19	40	54	49
4096	10	21	99	132	111	87
4096	30	9	50	105	103	68
4096	50	7	31	97	106	78

Table 5: Iteration counts for random data in the unit ball

remains convergent, but the speed of convergence may be rather slow. Our numerical investigations of this question employ right hand sides $f(\underline{x}_i)$, $i=1,2,\ldots,n$, that are chosen randomly from the uniform distribution on $[-1,1]$. The calculation of the algorithm is terminated when the residuals at the end of the k-th iteration satisfy the conditions

$$|r_i^{(k+1)}| \leq 10^{-8}, \qquad i=1,2,\ldots,n. \tag{3.26}$$

Table 5 reports the numbers of iterations that occur for the linear radial function $\phi(r)=r$, $r\geq 0$, for several values of d, n and q, when each data point $\underline{x}_i$ is picked at random from the unit ball $\{\underline{x} : \|\underline{x}\| \leq 1\} \subset \mathcal{R}^d$. We see that the approximations to the Lagrange functions are successful when d is only 2 and q is 30 or 50, fewer than 10 iterations being required when n is large. On the other hand, the number of iterations is roughly proportional to $\sqrt{n}$ for $d\geq 5$. It is remarkable that most of the iteration counts become smaller when d is increased from 20 to 40, which may happen because tiny distances between data points are less likely to occur in higher dimensions. Another unexpected feature of the last two columns of the table is that the choice $q=30$ is more efficient than $q=50$, especially if the preliminary work of calculating the coefficients $\zeta_{\ell j}$, $j \in \mathcal{L}_\ell$, $\ell=1,2,\ldots,n-1$, is taken into account. This remark receives further attention in the next section.

When d is large, however, we expect the points $\underline{x}_i$, $i=1,2,\ldots,n$, to lie on a low dimensional manifold in $\mathcal{R}^d$. Therefore we recall the parametric forms (2.5) for two and three dimensional manifolds, and we generate random points on these manifolds as before, by sampling the parameters ξ, η and ζ from the uniform distribution on $[0,2\pi]$. Further, the right hand sides $f(\underline{x}_i)$, $i=1,2,\ldots,n$, the termination condition (3.26), and the radial function $\phi(r)=r$, $r\geq 0$, are taken from the previous paragraph. Then our algorithm for solving the interpolation

n	q	$d=5$	$d=10$	$d=20$	$d=40$
256	10	18/25	18/26	16/25	17/25
256	30	10/14	9/16	9/16	9/16
256	50	8/11	6/11	6/11	7/11
1024	10	23/42	21/40	21/40	20/37
1024	30	12/22	10/24	10/21	9/21
1024	50	10/15	7/16	7/15	7/14
4096	10	26/74	25/65	24/57	24/58
4096	30	13/32	11/30	10/28	10/26
4096	50	10/21	8/21	8/17	7/17

Table 6: Iteration counts for data on manifolds in $\mathcal{R}^d$

equations (1.10) requires the numbers of iterations that are stated in Table 6, where the first and second numbers of each pair come from the two and three dimensional manifolds, respectively. We see that the two dimensional manifold entries are excellent, because, for the choices $q=30$ and $q=50$, there is little change in the iteration counts when n is increased. Moreover, by comparing Tables 5 and 6, we find that the three dimensional manifolds are also very helpful to the efficiency of the algorithm. Table 6 shows too that an increase in d hardly ever increases the number of iterations.

Another experiment that was tried was the replacement of $\phi(r)=r$ by the multiquadric $\phi(r)=(r^2+c^2)^{1/2}$ in the Table 5 calculations, where c has the value $(1/n)^{1/d}$. It happens, unfortunately, that more iterations are required to achieve the termination condition (3.26). For example, the entries of the $n=1024$ and $q=30$ row increase from 8, 30, 54, 53 and 44 to 8, 36, 106, 155 and 101, respectively. The possibility of reducing these iteration counts is considered later.

4. Discussion

It is mentioned in Section 1 that piecewise polynomial interpolation to values $f(\underline{x}_i)$, $i=1,2,\ldots,n$, of a function $f(\underline{x})$, $\underline{x}\in\mathcal{R}^d$, tends to be impracticable if d is large, especially if the data points $\underline{x}_i$, $i=1,2,\ldots,n$, are in general positions. Therefore the success of the radial basis function method that is reported in Section 2 is very welcome. We have found that, if the data points are on a low dimensional manifold in $\mathcal{R}^d$, then, without having to locate the manifold, the radial basis function method provides a particularly good approximation to f on the manifold automatically, and also fewer iterations are required by the algorithm of Section 3 for calculating the coefficients of the interpolant

$$s(\underline{x}) = \sum_{j=1}^{n} \lambda_j \, \phi(\|\underline{x}-\underline{x}_j\|) + \alpha, \qquad \underline{x} \in \mathcal{R}^d. \tag{4.1}$$

On the other hand, at first sight it is unattractive that the recommended choices of $\phi(r)$, $r \geq 0$, become unbounded as $r \to \infty$. We address some of the questions that arise from this situation.

It is elementary that there is no need for the growth in ϕ to impair the accuracy of the approximation $s \approx f$. This remark is justified usually by considering the linear radial function $\phi(r) = r$, $r \geq 0$, when $d = 1$. Then, as noted in Section 1, the function (4.1) becomes the piecewise linear interpolant to the data $f(\underline{x}_i)$, $i = 1, 2, \ldots, n$. Therefore one should certainly not take the view that the term $\lambda_j \phi(\|\underline{x}-\underline{x}_j\|)$, $\underline{x} \in \mathcal{R}^d$, indicates the magnitude of the contribution from $f(\underline{x}_j)$ to $s(\underline{x})$. Instead, this contribution is the product $f(\underline{x}_j)\chi_j(\underline{x})$, where χ_j is the Lagrange function of the interpolation method that satisfies the equations

$$\chi_j(\underline{x}_i) = \delta_{ij}, \qquad i = 1, 2, \ldots, n. \tag{4.2}$$

Therefore the magnitude of the contribution is known for the radial basis functions that have been mentioned, in the case when n is infinite and the data points are the vertices of the infinite grid $\mathcal{Z}^d$ (Buhmann, 1990). That work suggests that, for general positions of a finite number of data points, the factor $\chi_j(\underline{x})$ decays to zero rapidly as $\|\underline{x}-\underline{x}_j\|$ increases, if there are plenty of data, and if $\underline{x}$ is well inside the domain of the data points. Thus the contribution from $f(\underline{x}_j)$ to $s(\underline{x})$ tends to be negligible when $\phi(\|\underline{x}-\underline{x}_j\|)$ is very large.

Let n be huge, let the data be $f(\underline{x}_i)$, $i = 1, 2, \ldots, n$, and let an estimate of $s(\underline{x})$ be required for some fixed vector $\underline{x} \in \mathcal{R}^d$ that is well inside the domain of the data points, where s is the function of the form (4.1) that interpolates the given function values subject to $\sum_{j=1}^{n} \lambda_j = 0$, but we assume that the coefficients of s have not been calculated. The remarks of the last paragraph suggest the possibility of deriving an acceptable estimate of $s(\underline{x})$ from a subset of the data. For example, one may employ only the values of f at the 1000 interpolation points that are closest to $\underline{x}$, even if n is 10^6 or 10^9. Thus, apart from the identification of the 1000 points, the work of estimating $s(\underline{x})$ to prescribed accuracy may be bounded by an amount that is independent of n.

Similarly, the work of estimating the coefficient λ_j of the interpolant (4.1) may also be independent of n, if $\underline{x}_j$ is well inside the domain of the data points, j being an integer from $[1, n]$. Indeed, we deduce from the system (1.10) that λ_j is the sum

$$\lambda_j = \sum_{k=1}^{n} w_{jk} \, f(\underline{x}_k), \tag{4.3}$$

where w_{jk} is the (j,k)-th element of the inverse of the matrix of the system, so the value of w_{jk} depends on the positions of the data points but not on f. Moreover, the symmetry of Φ implies $w_{kj}=w_{jk}$, and, by letting $\underline{f}$ be the j-th coordinate vector in $\mathcal{R}^n$, it follows from expression (1.10) that the conditions (4.2) are satisfied by the function

$$\chi_j(\underline{x}) = \sum_{k=1}^{n} w_{kj}\,\phi(\|\underline{x}-\underline{x}_k\|) + w_{n+1\,j}, \qquad \underline{x}\in\mathcal{R}^d. \tag{4.4}$$

Thus, for general f, the k-th term of the sum (4.3) is just $f(\underline{x}_k)$ multiplied by the k-th coefficient of the Lagrange function χ_j. Now the decay of $\chi_j(\underline{x})$ as $\|\underline{x}-\underline{x}_j\|$ increases, mentioned in the second paragraph of this section, is inherited by the coefficients of χ_j. Therefore, when n is huge, equation (4.3) suggests that the contribution from $f(\underline{x}_k)$ to an estimate of λ_j may be negligible when $\underline{x}_k$ is far from $\underline{x}_j$.

This suggestion implies that the work of calculating all the coefficients λ_j, $j=1,2,\ldots,n$, of the interpolant (4.1) to prescribed accuracy should be much less than $\mathcal{O}(n^2)$, when n is sufficiently large. Therefore the matrix vector product $\Phi\underline{\delta}^{(k)}$ of the algorithm of Section 3, that occurs between equations (3.11) and (3.12), should be avoided. The components of the product are the values of the function

$$d^{(k)}(\underline{x}) = \sum_{j=1}^{n} \delta_j^{(k)}\phi(\|\underline{x}-\underline{x}_j\|), \qquad \underline{x}\in\mathcal{R}^d, \tag{4.5}$$

at the data points $\underline{x}_i$, $i=1,2,\ldots,n$. Fortunately, some massive breakthroughs in reducing the time of computing $d^{(k)}(\underline{x}_i)$, $i=1,2,\ldots,n$, have been developed from the fast multipole method of Greengard and Rokhlin (1987). Here the sum (4.5) is partitioned for each $\underline{x}$ 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 sum of the terms of the partition is approximated sufficiently accurately by an expression that depends on $\underline{x}$, and that can usually be calculated far more quickly than the original sum of the partition. Further, most of the approximations are employed many times as $\underline{x}$ runs through the set $\{\underline{x}_i : i=1,2,\ldots,n\}$. Details of these techniques for multiquadric radial functions are described by Cherrie, Beatson and Newsam (2002). They are highly efficient when the number of variables is two or three. We expect this success to extend to cases when d is large and the data lie on a low dimensional manifold in $\mathcal{R}^d$.

The algorithm of Section 3, unfortunately, is inconveniently slow on a Sun Sparc 10 workstation when there are more than 5000 data points, because of the omission of a fast method for estimating the product $\Phi\underline{\delta}^{(k)}$. The slowness is caused not only by the operations of the product, but also by the amount of disc storage that can be used efficiently when running a Fortran program. Indeed, for $n\geq 5000$, most of the elements of the lower triangular part of Φ have to be recalculated whenever they are required, which increases the computation time of the product by a multiple of $n^2 d$. On the other hand, if the total number of operations of a fast method is much less than n^2, then its storage requirements are much less than n^2 too. The effort of understanding and programming fast methods was not the only reason for excluding them from the software that produced our numerical results. It is also relevant that a fast method is worthwhile only if n is sufficiently large, which may require n to be huge for $d \geq 10$. Therefore we do not expect fast methods to reduce the work of the calculations of the last three columns of Tables 4 and 5.

It follows that any reductions in the iteration counts of Table 5 would be particularly welcome. There is room for improvement, because, as noted in Section 3, increasing q from 30 to 50 does not always decrease the number of iterations. Therefore another way of choosing the functions z_ℓ, $\ell = 1, 2, \ldots, n-1$, of formula (3.13) has been investigated by the author recently. The form

$$z_\ell(\underline{x}) = \sum_{j \in \mathcal{L}_\ell} \zeta_{\ell j} \, \phi(\|\underline{x} - \underline{x}_j\|), \qquad \underline{x} \in \mathcal{R}^d, \tag{4.6}$$

of expression (3.14) is retained, subject to $\sum_{j \in \mathcal{L}_\ell} \zeta_{\ell j} = 0$, where $\mathcal{L}_\ell$ is as before, ℓ being any integer from $[1, n-1]$, but the coefficients $\zeta_{\ell j}$, $j \in \mathcal{L}_\ell$, are defined by a method that is different from equations (3.23) and (3.24). Thus, if $q = 50$, for example, then it is possible for the functions z_ℓ, $\ell = 1, 2, \ldots, n-1$, to be identical to those that are chosen at present when $q = 30$. Further, in this example with $q = 50$, we allow only the $q = 30$ coefficients to be nonzero, and we try to improve the values of these coefficients by making use of the known positions of the interpolation points that become relevant when q is increased from 30 to 50. Specifically, because of the significance of the orthogonality conditions (3.22), we abandon the interpolation equations of expression (3.24), and instead we let the new $q = 30$ coefficients minimise the greatest value of the normalised scalar product $|\langle z_\ell, z \rangle_\phi| / \|z_\ell\|_\phi$, where z is any radial function with the centres $\underline{x}_i$, $i \in \mathcal{L}_\ell \backslash \{\ell\}$, that satisfies $\|z\|_\phi = 1$, the set $\mathcal{L}_\ell$ being the one that occurs in the case $q = 50$. The results of these investigations are reported in Section 5 of the paper that presents a full description of the algorithm (Faul, Goodsell and Powell, 2002). The main conclusion is that, although the new approach employs more interpolation points than before when calculating the nonzero coefficients of the function (4.6), the new values of theses coefficients are the same as the old ones.

A Fortran program that implements the algorithm is available from the author. It generates the coefficients of the interpolant (1.8) by solving the system (1.10), where $\phi(r)$, $r \geq 0$, is a linear or multiquadric radial function. There are no restrictions on the number of variables d, nor on the positions of the data points $\underline{x}_i \in \mathcal{R}^d$, $i = 1, 2, \ldots, n$, except that these points must all be different. The amount of work is of magnitude n^2, assuming that the number of iterations is independent of n, but, if there is not enough storage for the elements of the lower triangular part of the symmetric matrix Φ, then the multiplying factor includes a strong linear dependence on d. The lack of other methods for interpolation to values of functions of large numbers of variables should make the new software useful to many applications.

Acknowledgements

The algorithm of Section 3 was developed in collaboration with Anita Faul and George Goodsell. Further, George produced a Fortran 90 implementation of the algorithm, which was modified by the author to the Fortran 77 software that calculated the given numerical results. The contributions of Anita and George to the efficiency and convergence of the algorithm are of fundamental importance.

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.

A. Bejancu (1999), "Local accuracy for radial basis function interpolation on finite uniform grids", *J. Approx. Theory*, Vol. 99, pp. 242–257.

M.D. Buhmann (1990), "Multivariate cardinal interpolation with radial-basis functions", *Constr. Approx.*, Vol. 6, pp. 225–255.

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, G. Goodsell and M.J.D. Powell (2002), "A Krylov subspace algorithm for multiquadric interpolation in many dimensions", Report No. DAMTP 2002/NA5, 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.

L. Greengard and V. Rokhlin (1987), "A fast algorithm for particle simulations", *J. Comput. Phys.*, Vol. 73, pp. 325–348.

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.