RBF Collocation

C. T. Mouat and R. K. Beatson

Department of Mathematics & Statistics,
University of Canterbury,
Private Bag 4800, Christchurch, New Zealand.

Report Number: UCDMS2002/3 February 2002

Keywords: radial basis functions, domain decomposition, collocation, pde

RBF collocation

C. T. Mouat* and R. K. Beatson*

version of February 28, 2002

1 Introduction

In recent years radial basis function collocation has become a useful alternative to finite difference and finite element methods for solving elliptic partial differential equations. RBF collocation methods have been shown numerically (see for example [17]) and theoretically (see [14, 13]) to be very accurate even for a small number of collocation points. In application finite difference methods often have a low approximation order and consequently can require a large grid and considerable computation to obtain a sufficiently accurate solution. RBF collocation has been applied to linear elliptic PDEs in $\mathcal{R}^2$ and $\mathcal{R}^3$ [18], to time dependent problems [15, 16], and to non-linear problems [10].

In this paper we present new numerical results for RBF collocation. These results show that collocation with a basic function from the Matern class can be more accurate than collocation with the multiquadric basic function. Also, we present and implement an algorithm which solves linear and non-linear collocation equations with the multiquadric when N is large and $c < 2/\sqrt{N}$.

Section 2 briefly outlines RBF collocation and discusses difficulties with the method. These include poor conditioning and full matrices when using globally supported basic functions, and lower accuracy when using compactly supported basic functions. In Section

[1]Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

3 we give numerical results using a family of basic functions known as the Matern family. These numerical results show that this family is an effective alternative to the multiquadric basic function in many situations. Finally, in the last sections, we present a method which can be used to solve large collocation problems with the multiquadric basic function and $c < 2/\sqrt{N}$. Numerical experiments on linear PDEs show convergence to the solution for small enough values of c. It is hoped that in the future the algorithm will be able to be applied to larger values of c. This new algorithm combines the use of approximate cardinal functions and domain decomposition to iteratively find the solution of the collocation problem. Using approximate cardinal functions as a change of basis has been shown to be effective in the interpolation setting [2]. Previously solving a collocation system required $\mathcal{O}(N^3)$ operations (for globally supported Φ) and was not possible for large N. The method presented here solves the collocation system in $\mathcal{O}(N \log N)$ operations if $c < 2/\sqrt{N}$ and the PDE is suitable.

2 RBF collocation

This paper considers solving a suitable elliptic PDE of the form

$$\begin{aligned} Lu &= f \quad \text{in } \Omega \subset \mathcal{R}^d, \\ u &= g \quad \text{in } \partial\Omega, \end{aligned} \tag{1}$$

by radial basis function collocation. In (1) $f, g : \mathcal{R}^d \to \mathcal{R}$ are known and $\partial\Omega$ is the boundary of the region Ω. L is a differential operator and may be linear or non-linear. If L is non-linear a multilevel Newton iteration is required and a linearized system is solved at each level.

The unknown solution, u, to the PDE is approximated by a radial basis function, u_ϕ, of the form

$$u_\phi(\cdot) = p(\cdot) + \sum_{j=1}^{N} \lambda_j \Phi(\cdot - x_j). \tag{2}$$

Here $\lambda = [\lambda_1, \ldots, \lambda_N]^T$ are coefficients to be found, $p \in \pi_k^d$, and Φ is a basic function, such as the multiquadric. If L is time dependent then we let λ be a function of time and solve for $\lambda(t)$ at a finite number of discrete time steps. For more discussion on this case see [17]. For the moment assume L is time independent. Now for u_ϕ to satisfy the PDE (1) then

$$\begin{aligned} Lu_\phi(x) &= f(x), \quad x \in \Omega, \\ u_\phi(x) &= g(x), \quad x \in \partial\Omega. \end{aligned} \tag{3}$$

Clearly this cannot generally be achieved for every point in Ω. By choosing N distinct collocation points $X_I = \{x_1, \ldots, x_{N_I}\} \subset \Omega$ and $X_B = \{x_{N_I+1}, \ldots, x_N\} \subset \partial\Omega$ and ensuring (3) holds at these points we expect u_ϕ will be a good approximation to u. For the choice of u_ϕ in (2) the collocation equations are

$$Lp(x_i) + \sum_{j=1}^{N} \lambda_j L\Phi(x_i - x_j) = f_i, \quad i = 1, \ldots, N_I,$$

$$p(x_i) + \sum_{j=1}^{N} \lambda_j \Phi(x_i - x_j) = g_i, \quad i = N_I + 1, \ldots, N,$$

along with the side conditions

$$\sum_{j=1}^{N} \lambda_j q(x_j) = 0, \quad \text{for all } q \in \pi_k^d.$$

This leads to the equivalent matrix form

$$\begin{bmatrix} W_L & P_L \\ W_B & P_B \\ P^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}, \tag{4}$$

where

$$\begin{aligned} (W_L)_{ij} &= L\Phi(x_i - x_j), & x_i \in X_I, \;\; x_j \in X, \\ (W_B)_{i-N_I,j} &= \Phi(x_i - x_j), & x_i \in X_B, \;\; x_j \in X, \\ (P_L)_{ij} &= Lp_j(x_i), & x_i \in X_I, \\ (P_B)_{i-N_I,j} &= p_j(x_i), & x_i \in X_B, \end{aligned} \tag{5}$$

and $\{p_1, \ldots, p_{\dim(\pi_k^d)}\}$ forms a basis for π_k^d. The vector a consists of coefficients with respect to this basis. Solving this collocation system for the coefficients $[\lambda^T \;\; a^T]^T$, when N is large, is the emphasis of later sections of this paper. The strategy there is to precondition the collocation matrix

$$A = \begin{bmatrix} W_L & P_L \\ W_B & P_B \\ P^T & O \end{bmatrix}, \tag{6}$$

so that the preconditioned system is solved quickly using an iterative method. The collocation matrix, A, in (6) has not been proven to be non-singular but in [20] it was shown that finding a numerically singular matrix was very rare. The positioning of the centres has an effect on the accuracy of RBF collocation. However, to keep the discussion simpler, we only consider gridded centres.

Equation (2) is the form of the RBF approximation that was initially presented by Kansa [17]. This form is often called unsymmetric collocation due to the matrix in (6) being unsymmetric. An alternative approach [8], referred to as symmetric collocation, takes the form

$$u_\phi(\cdot) = p(\cdot) + \sum_{j=1}^{N_I} \lambda_j \widetilde{L}\Phi(\cdot - x_j) + \sum_{j=N_I+1}^{N} \lambda_j \Phi(\cdot - x_j), \tag{7}$$

where $\widetilde{L}$ is the operator L now applied to the second argument, x_j. Note that the absolute values of $\widetilde{L}\Phi(y-x)$ and $L\Phi(y-x)$ are equal for any x and y. For the choice of u_ϕ in (7) the collocation equations lead to the interpolation system

$$\begin{bmatrix} W_{L\widetilde{L}} & W_L & P_L \\ W_{\widetilde{L}}^T & W_B & P_B \\ P_L^T & P_B^T & O \end{bmatrix} \begin{bmatrix} \lambda \\ a \end{bmatrix} = \begin{bmatrix} f \\ g \\ 0 \end{bmatrix}. \tag{8}$$

The matrices in (8) are,

$$\begin{aligned}
(W_{L\widetilde{L}})_{ij} &= L\widetilde{L}\Phi(x_i - x_j), && x_i, x_j \in X_I, \\
(W_L)_{i,j-N_I} &= L\Phi(x_i - x_j), && x_i \in X_I, \;\; x_j \in X_B, \\
(W_{\widetilde{L}})_{i,j-N_I} &= \widetilde{L}\Phi(x_i - x_j), && x_i \in X_I, \;\; x_j \in X_B, \\
(W_B)_{i-N_I,j-N_I} &= \Phi(x_i - x_j), && x_i, x_j \in X_B,
\end{aligned} \tag{9}$$

and P_L and P_B are the same as in (4). The main advantage of this formulation is that it is provably non-singular (see [8, 22]). However the RBF in (7) is not as widely used as Kansa's original due to an extra application of L requiring that Φ be more differentiable. For nonlinear collocation using (7) also increases the complexity of the method. Some numerical results comparing the two approaches can be found in [8].

Both collocation systems are generally very badly conditioned which can restrict the use of RBF collocation to systems with only a few thousand centres. Theoretical results show that multiquadric interpolation becomes more accurate as the multiquadric parameter c increases [19]. A lot of numerical evidence agrees with this in the collocation setting. However, as c gets larger the graph of the basic function becomes flatter and this leads to bad conditioning. Thus as the accuracy of the approximation increases then often so does the ill-conditioning. Various techniques have been used with mixed success to combat this problem (see for example [18]).

The problems associated with using globally supported basic functions have led to the use of compactly supported basic functions such as the Wendland functions [21]. If the support is small then matrix-vector multiplies can be calculated in $\mathcal{O}(N)$ operations. The problem with compactly supported basic functions is that good approximations to the solution are only obtained when the support is large. For accurate results the sparcity of the matrix is lost. A multilevel approach with smoothing can improve the accuracy of the RBF approximation [7] but multiquadric basic functions are usually more accurate.

3 Collocation with Matern basic functions

Traditionally multiquadric or compactly supported basic functions are the preferred choice for RBF collocation. Numerical evidence has shown good results with these choices of basic functions for various types of problems. Other alternatives that are common in the RBF interpolation setting can be restricted in their use for collocation. For example, the Laplacian of the thin-plate spline is

$$\Delta\Phi(x) = 4\log(\|x\|) + 4,$$

which has a discontinuity at zero. The Laplacian of the exponential basic function also has a discontinuity at zero. This makes the use of the thin-plate spline and exponential limited in RBF collocation.

Due to the conditioning problems associated with the multiquadric we consider the use of alternative basic functions for RBF collocation. This section presents numerical results for some simple PDEs using the Matern family as basic functions. The Matern family is given by

$$\phi_\nu(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}(cr)^\nu K_\nu(cr), \tag{10}$$

where K_ν is a modified Bessel function of order $\nu > 0$ (note that ν is also a smoothness parameter) and $c > 0$. If n is a nonnegative integer then (10) simplifies to

$$\phi_{n+1/2}(r) = \frac{\exp(-cr)(cr)^n}{(2n-1)!!}\sum_{k=0}^{n}\frac{(n+k)!}{k!(n-k)!(2cr)^k}.$$

Some examples for various values of ν are:

$$\begin{aligned}
&\nu = 1/2, \quad \phi(r) = \exp(-cr),\\
&\nu = 1, \quad\;\; \phi(r) = crK_1(cr),\\
&\nu = 3/2, \quad \phi(r) = (1+cr)\exp(-cr),\\
&\nu = 5/2, \quad \phi(r) = (1+cr+c^2r^2/3)\exp(-cr).
\end{aligned} \tag{11}$$

Although we only consider unsymmetric collocation here the motivation behind the use of the Matern class comes from the results of Franke and Schaback [14, 13] in the symmetric

collocation setting. They show that for a PDE of order m the L_∞ approximation order for RBF collocation with a Matern basic function will be $\nu - m$. Note that this result is for solutions u in the "native space" of Φ. A complete review of the work of Franke and Schaback is beyond the scope of this thesis but the reader is referred to their papers [14, 13].

Tables 1 and 2 contain condition numbers of the collocation matrix (6) and relative error results for the PDE

$$\begin{aligned} \Delta u &= 32\cos(4x_1 + 4x_2), && (x_1,\ x_2) \in \Omega, \\ u &= \cos(4x_1 + 4x_2), && (x_1,\ x_2) \in \partial\Omega, \end{aligned}$$

where Ω is the unit square. The relative error is $\|s - u\|_2/\|u\|_2$ where s is the values of the RBF and u the values of the true solution evaluated on a uniform grid of size $(2\sqrt{N} - 1) \times (2\sqrt{N} - 1)$. The basic functions we compare are the multiquadric and the Matern, $\nu = 9/2$, function.

It is clear from the tables that as the basic function becomes flatter the condition number increases for a fixed set size. In the case of the multiquadric this corresponds to c increasing, whereas for the Matern function this corresponds to a decrease in c.

Table 1 shows results for centres on a uniform grid in $[0,\ 1]^2$. The smallest relative error for the Matern function is about 9 times smaller then the smallest relative error for the multiquadric. However, both these experiments have condition numbers greater than 10^{20}. If we look at experiments with condition numbers that are about 10^{16} or less then the difference between the basic functions is even more dramatic. The best results are then approximately 1.7×10^{-5} and 4×10^{-7} for the multiquadric and Matern functions respectively. The error for the Matern function is about 40 times smaller than the error for the multiquadric!

The same experiments were repeated on a grid of shifted Chebychev nodes in $[0,\ 1]^2$. The results are in Table 2. The errors for these trials were as low as 1.13×10^{-8} for Matern collocation on 4225 centres. Overall, for this PDE, RBF collocation with the Matern basic function was more accurate than RBF collocation with the multiquadric especially for

large N. Also collocation on the Chebychev grid was more accurate than collocation on the uniform grid.

Number of centres N	Multiquadric			Matern, $\nu = 9/2$		
	c	relative error	condition number	c	relative error	condition number
9×9	15/9	7.552(-5)	6.555(17)	0.5	1.056(-3)	1.694(16)
	13/9	1.217(-4)	1.038(17)	1.0	1.079(-3)	3.069(13)
	11/9	2.582(-4)	2.037(15)	1.5	1.108(-3)	9.406(11)
	9/9	5.684(-4)	2.877(13)	2.0	1.163(-3)	9.117(10)
	7/9	1.250(-3)	2.447(11)	2.5	1.251(-3)	1.601(10)
17×17	15/17	2.095(-6)	1.492(19)	0.5	6.860(-4)	6.704(18)
	13/17	5.656(-6)	3.610(19)	1.0	4.488(-5)	1.603(17)
	11/17	1.854(-5)	5.778(18)	1.5	4.703(-5)	3.023(15)
	9/17	6.131(-5)	2.378(16)	2.0	5.175(-5)	3.109(14)
	7/17	2.152(-4)	4.151(13)	2.5	5.796(-5)	5.518(13)
33×33	15/33	1.895(-6)	2.594(20)	2.0	1.628(-6)	1.338(19)
	13/33	9.995(-7)	4.394(20)	2.5	1.952(-6)	1.484(17)
	11/33	3.345(-6)	9.664(20)	3.0	2.266(-6)	3.162(16)
	9/33	1.397(-5)	2.063(18)	3.5	2.684(-6)	9.402(15)
	7/33	5.648(-5)	1.397(15)	4.0	3.232(-6)	3.188(15)
65×65	15/65	1.536(-6)	1.862(21)	3.0	2.139(-7)	3.682(20)
	13/65	1.099(-6)	3.087(21)	4.0	1.112(-7)	1.472(20)
	11/65	1.423(-6)	8.629(20)	5.0	1.419(-7)	3.792(18)
	9/65	4.170(-6)	7.398(20)	6.0	2.290(-7)	8.410(17)
	7/65	1.697(-5)	2.086(16)	7.0	3.724(-7)	5.944(16)

Table 1: Radial basis function collocation of the Poisson equation with solution $\cos(4x_1 + 4x_2)$. Uniform grid.

Number of centres N	Multiquadric c	Multiquadric relative error	Multiquadric condition number	Matern, $\nu = 9/2$ c	Matern, $\nu = 9/2$ relative error	Matern, $\nu = 9/2$ condition number
9×9	15/9	3.026(-5)	4.737(17)	0.5	3.966(-4)	1.775(16)
	13/9	4.962(-5)	2.614(16)	1.0	4.164(-4)	3.464(13)
	11/9	1.038(-4)	6.360(14)	1.5	4.342(-4)	1.085(12)
	9/9	2.307(-4)	1.102(13)	2.0	4.634(-4)	1.059(11)
	7/9	5.189(-4)	1.305(11)	2.5	5.066(-4)	1.864(10)
17×17	15/17	4.469(-5)	1.454(19)	1.0	4.136(-6)	4.806(18)
	13/17	4.756(-7)	7.760(18)	2.0	2.498(-6)	4.190(15)
	11/17	7.822(-7)	5.132(18)	3.0	3.370(-6)	1.732(14)
	9/17	2.924(-6)	9.153(16)	4.0	4.865(-6)	1.725(13)
	7/17	1.344(-5)	5.349(14)	5.0	7.407(-6)	2.721(12)
33×33	11/33	6.382(-6)	9.656(19)	4.0	1.222(-7)	1.957(20)
	9/33	2.256(-5)	6.774(19)	5.0	1.554(-8)	2.005(18)
	7/33	1.082(-6)	3.598(19)	6.0	2.389(-8)	1.843(17)
	5/33	4.329(-6)	6.523(17)	7.0	3.878(-8)	3.591(16)
	3/33	1.549(-5)	7.397(13)	8.0	6.210(-8)	1.226(16)
65×65	5/65	9.659(-7)	1.204(22)	15	1.213(-8)	6.011(20)
	4/65	7.014(-7)	4.365(20)	20	1.127(-8)	3.828(18)
	3/65	2.533(-6)	1.893(20)	25	4.414(-8)	5.882(17)
	2/65	1.105(-4)	4.786(16)	30	2.111(-7)	1.131(17)
	1/65	4.847(-3)	3.162(12)	35	9.722(-7)	3.879(16)

Table 2: Radial basis function collocation of the Poisson equation with solution $\cos(4x_1 + 4x_2)$. Chebychev grid.

4 Solving the collocation system for large N

For globally supported basic functions directly solving the collocation system (4) requires $\mathcal{O}(N^3)$ operations and $\mathcal{O}(N^2)$ storage without using any customised method. This section presents a new algorithm for solving this system in $\mathcal{O}(N \log N)$ operations and $\mathcal{O}(N)$ storage with the multiquadric basic function. The algorithm uses a change of basis preconditioner in conjunction with domain decomposition and a fast matrix-vector multiply. The greatest computational cost at each iteration is at least one matrix-vector multiply. Fast matrix-vector product algorithms allow this to be achieved in $\mathcal{O}(N \log N)$ operations using a suitable fast evaluation code. These fast evaluation codes exist for a variety of functions [1, 3, 5].

4.1 Domain decomposition

This subsection considers a domain decomposition algorithm for centres separated into two domains. Without loss of generality refer to $X_I = \{x_1, \ldots, x_{N_I}\}$ as good points and $X_B = \{x_{N_I+1}, \ldots, x_N\}$ as bad points. Also assume that $N_I >> |X_B| =: N_B$. Then if an interpolant, s, is of the form

$$s(\cdot) = \sum_{j=1}^{N} \lambda_j \Psi(\cdot - x_j),$$

where the coefficients λ_j are to be found then the interpolation matrix is $B_{ij} = \Psi(x_i - x_j)$. This can be split into the form

$$B = \begin{bmatrix} B_{II} & B_{IB} \\ B_{BI} & B_{BB} \end{bmatrix}. \tag{12}$$

In (12) $B_{jk},\ j,k \in \{I,B\}$ has size $N_j \times N_k$ and is the matrix from evaluating the Ψs centred at X_k, at points in X_j. Now applying a simple domain decomposition algorithm to this system we can iteratively obtain a solution. This method is given by Algorithm 4.1. The notation r_B, r_I refers to the residuals restricted to centres X_B and X_I respectively.

Algorithm 4.1 $domain - decomposition(X, f, N_I)$

SETUP

1. Create a set, X_I, of good points and a set, X_B of bad points
2. Form Ψ elements for each point x in X
3. $r \leftarrow f$ and $s \leftarrow 0$

ITERATIVE SOLUTION

1. **while** $\|r\| > \epsilon$
2. Solve for the coefficients μ of a bad point approximation via direct or approximate solutions of $B_{BB}\mu = r_B$
3. $s_{\text{bad}} \leftarrow \sum_{j:x_j \in X_B} \mu_j \Psi_j$
4. Evaluate $r_I = r_I - s_{\text{bad}}(X_I)$
5. Solve for the coefficients μ of a good point approximation via approximate solution of $B_{II}\mu = r_I$
6. $s_{\text{good}} \leftarrow \sum_{j:x_j \in X_I} \mu_j \Psi_j$
7. Update the RBF $s = s + s_{\text{bad}} + s_{\text{good}}$
8. Update the residual $r = f - s(X)$
9. **end while**

At the beginning of each iteration coefficients μ are found so that

$$\sum_{j:x_j \in X_B} \mu_j \Psi_j(x_i) = r_i, \quad x_i \in X_B.$$

Because N_B is small compared to N_I this is relatively efficient to solve. The residuals are then updated and a similar system on the good points is solved. Although N_I is large we can solve the system on the good points efficiently by GMRES if the eigenvalues of B_{II} are sufficiently clustered. Each GMRES iteration will require the computational cost of a matrix-vector multiply. In our experience an exact solution at step 5 is not required. Instead reducing the residual by a few orders of magnitude will suffice. If B_{II} is an approximation to the identity then most off diagonal elements will be near zero. An approximation to B_{II} can easily be found by retaining only a small number, say σ, of the largest magnitude entries per column. A matrix-vector product will then only require $\mathcal{O}(\sigma N)$ operations instead of the $\mathcal{O}(N \log N)$ required using a fast matrix-vector code. Numerical evidence shows that this approximation increases the number of outer iterations by less then four times but significantly decreases the total number of $\mathcal{O}(N \log N)$ matrix-vector products (which is the main computational cost of the algorithm).

The final step is to update the residual which can be achieved in $\mathcal{O}(N \log N)$ operations. Note that all matrix-vector multiplies will be $\mathcal{O}(N \log N)$ only if a suitable fast evaluation algorithm exists for the basic functions Φ and $L\Phi$. For example, if L is the Laplacian and Φ the multiquadric then

$$L\Phi(\cdot) = \frac{\|\cdot\|^2 + 2c^2}{(\|\cdot\|^2 + c^2)^{3/2}} = \frac{1}{(\|\cdot\|^2 + c^2)^{1/2}} + \frac{c^2}{(\|\cdot\|^2 + c^2)^{3/2}},$$

which is a combination of two members of the multiquadric family. Fast evaluators are available for functions of this type [3].

Algorithm 4.1 should be modified to include a coarse grid correction at each iteration. We usually take the number of points in the coarse grid to be about N_B.

4.2 Approximate cardinal functions

In the previous section it was assumed that the matrix B_{II} had clustered eigenvalues. In this section we achieve this by forming Ψ elements as approximate cardinal functions. We also explain why this approach doesn't work for large values of the multiquadric parameter c. Using approximate cardinal functions as a change of basis has been shown to be effective

in the interpolation setting [2, 4]. The main difference in the collocation case is that there are different operators on the interior and on the boundary. If our aim was for B to be a good approximation to the identity, as in the interpolation case, then the Ψ_j's would be of the form,

$$\begin{aligned} \Psi_j(x_i) &\approx 0, \quad x_i \in X_B, \\ L\Psi_j(x_i) &\approx 0, \quad x_i \in X_I, \end{aligned}$$

along with one of the constraints,

$$\begin{aligned} L\Psi_j(x_j) &= 1, \text{ if } \quad x_j \in X_I, \\ \Psi_j(x_j) &= 1, \text{ if } \quad x_j \in X_B. \end{aligned}$$

In our experience forming approximate cardinal functions to satisfy these conditions is difficult. Instead we form approximate cardinal functions which ensure B_{II} is a good approximation to the identity and use the domain decomposition approach given in Algorithm 4.1. The bad points are the boundary points and the good points are the interior points. For a uniform distribution of points in $\mathcal{R}^2$ the number of boundary points, N_B, is proportional to $N^{1/2}$ so direct solution of a linear system on these points requires $\mathcal{O}(N^{3/2})$ operations. Calculating the LU factorisation to B_{BB} as part of the setup means this cost is only incurred once. Subsequent use of this LU decomposition to solve a system requires $\mathcal{O}(N_B^2) = \mathcal{O}(N)$ operations.

Each Ψ element is of the form

$$\Psi_j(\cdot) = p_j(\cdot) + \sum_{i \in \mathcal{S}_j} \lambda_{ji} \Phi(\cdot - x_i),$$

where the set $\mathcal{S}_j$ is often a set of indices of the nearest β points to x_j. For interior points we construct Ψ elements so that

$$\begin{aligned} L\Psi_j(x_j) &= 1, \\ L\Psi_j(x) &= \mathcal{O}(\|x - x_j\|^{-3}) \text{ as } \|x - x_j\| \to \infty. \end{aligned}$$

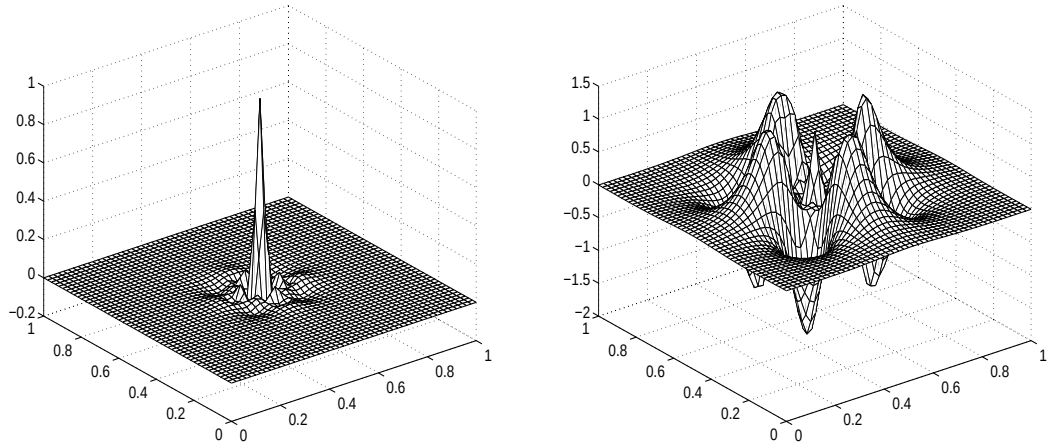
Approximate cardinal functions of this type are referred to as decay element approximate cardinal functions and are found by a constrained least squares problem as mentioned in [2].

The set X_j is defined to be the centres in X such that $x_i \in X_j$ if and only if $i \in \mathcal{S}_j$. For boundary points use a pure local approach of the form

$$\begin{aligned} \Psi_j(x_j) &= 1 \\ \Psi_j(x_i) &= 0, \quad x_i \in X_j \cap X_B, \\ L\Psi_j(x_i) &= 0, \quad x_i \in X_j \cap X_I. \end{aligned}$$

The pure local approximate cardinal functions are found by solving a collocation system on $|\mathcal{S}_j|$ nodes for each j.

In our experience we have noticed that creating approximate cardinal functions is only effective for $c < 2/\sqrt{N}$ (if the centres are a uniform grid in $[0,\ 1]^2$). In Figure 1 approximate cardinal functions formed using a decaying strategy for two different values of c are compared. Clearly the Ψ element formed with the larger value of c is not a very good approximation to a cardinal function. The required rate of decay is not achieved until further away from the centre of the Ψ element. An explanation for this is the regions of validity of the far field expansions. Consider finding a Ψ centred at x_j and based on centres with maximum distance H from x_j. The Ψ element will decay in the region of validity of the far field expansion of the cluster. This expansion is given in [3] and is valid outside the circle $\|x - x_j\| = \sqrt{H^2 + c^2}$. If c is large then the radius of this circle increases and the region of validity is further from x_j (see Figure 2) and thus the decay of Ψ occurs further away.



(a) A Ψ function formed with $c = 2/33$.

(b) A Ψ function formed with $c = 4/33$.

Figure 1: Ψ elements based on fifty local centres for two different values of c. Notice that the Ψ function decays quicker with the smaller value of c.

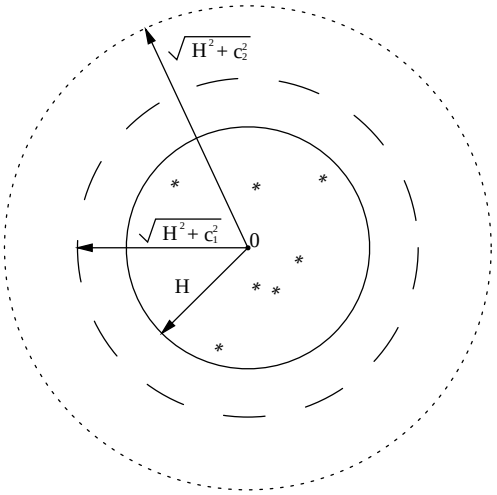


Figure 2: Far field expansion regions of validity for centres inside the circle $\| \cdot \| \leq H$ and multiquadric parameters c_1, c_2 with $c_2 > c_1$. The region of validity corresponding to c_2 is outside the dotted circle and for c_1 is outside the dashed circle.

5 Numerical results

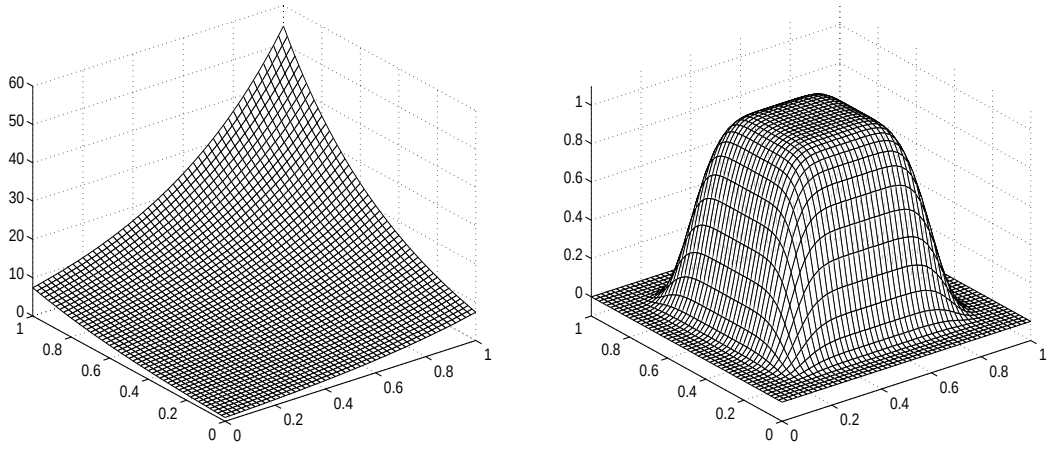
In this section we present numerical results for RBF collocation of linear PDEs of the form (1). These results show good convergence of the algorithm when the multiquadric parameter c is suitably small and constant. All numerical experiments are in the domain $[0,\ 1]^2$ with the collocation nodes forming an $n \times n$ grid. The iterations are stopped once the relative 2-norm residual is less than 10^{-8}.

To initially try this method we consider solving Poissons equation in $\mathcal{R}^2$ with the solutions

$$f_1(x_1, x_2) = \exp\left(2x_1 + 2x_2\right), \tag{13}$$

$$f_2(x_1, x_2) = \exp\left(-1000((x_1 - 1/2)^6 + (x_2 - 1/2)^6)\right). \tag{14}$$

RBF collocation solutions for these two PDEs can be found in Figure 3. The results from



(a) Exact solution is f_1.

(b) Exact solution is f_2.

Figure 3: RBF collocation on a 33×33 grid of centres in $[0,\ 1]^2$ with $c = 2/33$.

applying Algorithm 4.1 to f_1 and f_2 are in Tables 3 and 4 respectively. The algorithm was applied using both exact matrix-vector products and approximate matrix-vector products at step 5. We will refer to these different implementations as Algorithm 4.1(a) and Al-

gorithm 4.1(b) respectively. For both implementations exact matrix-vector products are always used at step 7. The "matrix-vector" column in the tables give the total number of exact matrix-vectors calculated to find the solution. For Algorithm 4.1(b) the total number of exact matrix-vector products is equal to the number of outer iterations. The "2-norm residual" column is the relative 2-norm residual $\|A\lambda - \bar{f}\|_2/\|\bar{f}\|_2$ where $\bar{f}$ is the right hand side vector $[f^T \quad g^T]^T$ in (4). The tables show that the algorithm converges for both small and large values of N. As expected, Algorithm 4.1(b) requires more outer iterations to converge but the total number of exact matrix-vector products is reduced. This is a sizeable computational saving for large N.

Overall the number of outer iterations remained fairly stationary for Algorithm 4.1(a). When approximate matrix-vectors were used the number of outer iterations increased slightly but not dramatically as N increased. Thus it would be feasible to solve even larger systems using this algorithm.

From these experiments we can conclude that the algorithm will at least work on some simple PDEs when c is small. Usually c is required to be large for higher accuracy but in the case of f_2 we noticed that $c = 2/\sqrt{N}$ was nearly optimal for small data sets and using a `Matlab` $\backslash$ operator to solve the systems. Carlson and Foley [6] suggest that a small shape parameter will be more accurate if the function values vary rapidly. The algorithm presented here may therefore be more applicable for solutions of this type.

A suitably modified algorithm has shown promising results for the nonlinear PDE given in [11, 10].

References

[1] R. K. Beatson and E. Chacko, Fast evaluation of radial basis functions: A multivariate momentary evaluation scheme, to appear in *Curve and Surface Fitting: Saint Malo 1999*, A. Cohen, C. Rabut and L.L. Schumaker (eds), Vanderbilt Univ. Press, Nashville (2000).

N	N_C	Exact matrix-vectors			Approximate matrix-vectors		
		Outer iterations	Matrix vectors	2-norm residual	Outer iterations	σ	2-norm residual
289	64	8	48	9.323(-9)	9	43.2	8.996(-9)
1089	121	8	48	1.413(-9)	14	44.6	6.745(-9)
4225	400	7	42	7.051(-9)	20	47.4	7.004(-9)
16641	625	9	54	2.409(-9)	27	48.2	8.539(-9)

Table 3: Results from Algorithm 4.1 on function f_1. N_C is the number of coarse grid points and σ is the average number of non-zero elements per column in the approximation to B.

N	N_C	Exact matrix-vectors			Approximate matrix-vectors		
		Outer iterations	Matrix vectors	2-norm residual	Outer iterations	σ	2-norm residual
289	64	9	54	1.940(-9)	9	43.2	9.313(-9)
1089	121	7	42	1.576(-9)	16	44.6	8.487(-9)
4225	400	6	36	2.313(-9)	22	47.4	5.396(-9)
16641	625	7	42	2.174(-9)	28	48.2	6.754(-9)

Table 4: Results from Algorithm 4.1 on function f_1. N_C is the number of coarse grid points and σ is the average number of non-zero elements per column in the approximation to B.

[2] R. K. Beatson, J. B. Cherrie and C. T. Mouat, Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration, *Advances in Computational Mathematics*, **11** (1999), 253-270.

[3] R. K. Beatson, J. B. Cherrie and G. N. Newsam, Fast evaluation of radial basis functions: Methods for generalised multiquadrics in $\mathcal{R}^n$. Manuscript (2001).

[4] R. K. Beatson and C. T. Mouat, Fast Kriging, *University of Canterbury Research Report number UCDMS2000/2*, (2000).

[5] R. K. Beatson and G. N. Newsam, Fast evaluation of radial basis functions: I, *Computers and Mathematics with Applications*, **24**, **No. 12** (1992), 7-19.

[6] R. E. Carlson and T. A. Foley, The parameter R^2 in multiquadric interpolation, *Computers and Mathematics with Applications*, **21** (1991), 29-42.

[7] G. E. Fasshauer, On the numerical solution of differential equations with radial basis functions. *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia, and D. W. Pepper (eds), WIT Press, (1999), 291-300.

[8] G. E. Fasshauer, Solving partial differential equations by collocation with radial basis functions. *Surface Fitting and Multiresolution Methods*, A. Le Mehaute, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press, (1997), 131-138.

[9] G. E. Fasshauer and J. W. Jerome, Multistep approximation algorithms: improved convergence rates through postconditioning with smoothing kernels. *Advances in Computational Mathematics*, **10** (1999), 1-27.

[10] G. E. Fasshauer, Nonsymmetric multilevel RBF collocation within an operator newton framework for nonlinear PDEs, to appear in *Trends in Approximation Theory*.

[11] G. E. Fasshauer, E. C. Gartland and J. W. Jerome, Newton iteration for partial differential equations and the approximation of the identity, *Numerical Algorithms*, **25** (2000), 181-195.

[12] R. Franke, Scattered data interpolation: tests of some methods, *Mathematics of Computation*, **38** (1982), 181-200.

[13] C. Franke and R. Schaback, Convergence order estimates of meshless collocation methods using radial basis functions, *Advances in Computational Mathematics*, **8** (1998), 381-399.

[14] C. Franke and R. Schaback, Solving partial differential equations by collocation using radial basis functions, *Applied Mathematics and Computation*, **93** (1998), 73-82.

[15] Y. C. Hon, A RBFs method for solving options pricing model, *Proceedings of Advances in Scientific Computing and Modeling*, Alicante, Spain (1998), 210-230.

[16] Y. C. Hon and Z. Wu, Error estimation on using radial basis functions for solving option pricing models. Manuscript.

[17] E. J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics-II, *Computers and Mathematics with Applications*, **19** (1990), 147-161.

[18] E. J. Kansa and Y. C. Hon, Circumventing the ill-conditioining problem with multiquadric radial basis functions: applications to elliptic partial differential equations, *Computers and Mathematics with Applications*, **39** (2000), 123-137.

[19] W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, *Computers and Mathematics with Applications*, **24** (1992), 121-138.

[20] R. Schaback and Y. C. Hon, On unsymmetric collocation by radial basis functions, *Journal of Applied Mathematics and Computation*, **119** (2001), 177-186.

[21] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in Computational Mathematics*, **4** (1995), 389-396.

[22] Z. Wu, Hermite-Birkhoff interpolation of scattered data by radial basis functions, *Approximation Theory and its Applications*, **8** (1992), 1-10.