On Approximate Cardinal Preconditioning Methods for Solving PDEs with Radial Basis Functions

Damian Brown

Department of Mathematics, University of Leicester, University Road, Leicester, LE1 7RH UK.

Leevan Ling

Department of Mathematics, City University of Hong Kong, 83, Tat Chee Avenue, Kowloon, Hong Kong.

Edward Kansa

Department of Mechanical and Aeronautical Engineering, One Shields Ave., University of California, Davis, Davis, CA 95616 USA.

Jeremy Levesley

Department of Mathematics, University of Leicester, University Road, Leicester, LE1 7RH UK.

Abstract

The approximate cardinal basis function (ACBF) preconditioning technique has been used to solve partial differential equations (PDEs) with radial basis functions (RBFs). In [31], a preconditioning scheme that is based upon constructing the least-squares approximate cardinal basis function from linear combinations of the RBF-PDE matrix elements has shown very attractive numerical results. This preconditioning technique is sufficiently general that it can be easily applied to many differential operators.

In this paper, we review the ACBF preconditioning techniques previously used for interpolation problems and investigate a class of preconditioners based on the one proposed in [31] when a cardinality condition is enforced on different subsets. We numerically compare the ACBF preconditioners on several numerical examples of Poisson's, modified Helmholtz and Helmholtz equations, as well as a diffusion equation and discuss their performance.

Key words: radial basis function, partial differential equation, preconditioner, cardinal basis function

Hardy's multiquadric	$\phi(r) = \sqrt{r^2 + c^2}$,
Inverse multiquadric	$\phi(r) = 1/\sqrt{r^2 + c^2}$,
Gaussian spline	$\phi(r) = e^{-cr^2}$,

Table 1
Examples of infinitely smooth RBFs.

---

1 Introduction

In the last 20 years many researchers have shown interest in mesh-free radial basis functions (RBFs) methods. RBFs have been used for interpolation problems [6,8–10,36] as well as for numerically solving partial differential equations (PDEs) [22–26,28]. In addition, Chen [12–14] investigated the use of distance functions that are related to RBFs to solve various PDE systems. The method is attractive not only because of its spectral accuracy [15,18,19,33,34,40,41] when using Gaussians or multiquadrics but its ability to work on scattered data without using any mesh. The mesh generation problem over irregularly shaped domains is often in excess of 70% of the total computational cost.

The price for this increased accuracy is usually ill-conditioning of the associated linear systems that need to be solved: the "uncertainty relation" shown by Schaback [37–39] is that better conditioning is associated with poorer accuracy and worse conditioning is associated with improved accuracy. Different approaches have already been proposed to overcome the difficulties, for instance, see [1,3,17,21,27,29,31].

The idea of RBFs is to use linear combinations of translates of a basis function $\phi(r)$ of one variable, expanded about the given scattered "data centers" $\vec{x}_j \in \mathbb{R}^d$, $j = 1, \ldots, N$ to approximate an unknown function by

$$s(\vec{x}) = \sum_{j=1}^{N} \lambda_j \phi(\|\vec{x} - \vec{x}_j\|), \tag{1}$$

where $\vec{x} \in \mathbb{R}^d$ and $\|\cdot\|$ is the Euclidean norm.

In this paper, we focus upon the asymmetric formulation [25,26] of PDEs using infinitely smooth RBFs; some examples are given in Table 1. Consider

a boundary value problem of the form

$$\begin{aligned} &\mathcal{L}u = f(\vec{x}) \text{ in } \Omega \subset \mathbb{R}^d, \\ &\mathcal{B}u = g(\vec{x}) \text{ on } \partial\Omega, \end{aligned} \tag{2}$$

where $\partial\Omega$ denotes the boundary of the domain Ω. Let $\mathcal{L}$ be the differential operator that operates on the interior, and $\mathcal{B}$ be an operator that specifies the boundary conditions of Dirichlet, Neumann or mixed type. Both f and g are given functions that map $\mathbb{R}^d \to \mathbb{R}$.

The unknown PDE solution u is approximated by RBFs in the form of (1). We assume the collocation points are arranged in such a way that the first N_I points and the last N_B points are in Ω and on $\partial\Omega$, respectively. If one collocates at the data centers, the collocation conditions of (2) are

$$\begin{aligned} f_i = f(\vec{x}_i) &= \sum_{j=1}^{N} \lambda_j \mathcal{L}\phi_j(\vec{x}_i), \text{ for } i = 1, \ldots, N_I, \\ g_i = g(\vec{x}_i) &= \sum_{j=1}^{N} \lambda_j \mathcal{B}\phi_j(\vec{x}_i), \text{ for } i = N_I{+}1, \ldots, N, \end{aligned} \tag{3}$$

where $\phi_i(\cdot) = \phi(\|\cdot - \vec{x}_i\|)$. The equations in (3) can be summarized in a system of equations for the unknown coefficients λ_j,

$$A\vec{\lambda} = \vec{f}, \tag{4}$$

where $\vec{\lambda} = (\lambda_1, \ldots, \lambda_N)^T$, and $\vec{f} = (f_1, \ldots, f_{N_I}, g_{N_I+1}, \ldots, g_N)^T$, $A_{ij} = \mathcal{L}\phi(\vec{x}_i - \vec{x}_j)$ for $i = 1, \ldots, N_I$, $A_{ij} = \mathcal{B}\phi(\vec{x}_i - \vec{x}_j)$ for $i = N_I{+}1, \ldots, N$, and $j = 1, \ldots, N$ in both cases.

The matrix system given in (4) is generally non-symmetric and full. This system of equations is known to be ill-conditioned, even for moderate N. This ill-conditioning worsens with N or with a flat RBF (e.g. the multiquadric with large shape parameter c). Although some very rare combinations of data center arrangements and c can produce a singular matrix, the singularity can be removed by perturbing either the value of c or the data centers; in the event of a singular matrix, preconditioning is a futile effort.

The outline of this paper is as follows: we begin by reviewing the existing approximate cardinal function preconditioning methods for the interpolation problems and the RBF-PDE problems in Section 2. We present the estimated number of flops for each preconditioner, CPU time can then be computed by multiplying this by the flop rate of the computer used. In Section 3, we

review how the fast multipole method can be employed to reduce the cost of solving PDEs with MQ-RBF. Then, we present the numerical performance of the different ACBF preconditioners when applied to the Poisson's, modified Helmholtz and Helmholtz equations, as well as a diffusion equation, in Section 4. We summarize the results in Section 5.

2 Approximate cardinal function preconditioning methods

Given centers $X = \{\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_n\}$, a cardinal function for interpolation problems associated with $\vec{x}_i$ has the form

$$\psi_i(\cdot) = \sum_{j=1}^{N} \tilde{w}_{ji} \phi(\| \cdot - \vec{x}_j \|), \tag{5}$$

where $\psi_i(\vec{x}_i) = 1$, and $\psi_i(\vec{x}_j) = 0$ for $j = 1, \ldots, N,\ j \neq i$. In such a case, the matrix $\tilde{W}$ with elements $\tilde{w}_{ji}$ would be the inverse of the RBF interpolation matrix on the data set X.

We compute a preconditioner W such that

$$WA\vec{\lambda} = W\vec{f}$$

is easier to solve in terms of GMRES iterations than Equation 4.

In all cases discussed in this section, the resulting preconditioners are sparse: there are only σ nonzero elements in each row. Multiplying the preconditioners by a vector can be performed in $\mathcal{O}(\sigma N)$ flops.

2.1 Approximate cardinal basis functions based on solving local interpolation problems, minimal-ACBF

Fix $\sigma \ll N$ and let $\mathcal{S}_i = [s_i(1), \ldots, s_i(\sigma)]$ be a subset of the index $[1, 2, \ldots, N]$ associated with the center $\vec{x}_i$. Suppose the expression (5) is formed by a relatively small set of RBFs instead of the whole set X, giving an approximate cardinal basis function (ACBF). Then, we ensure that

$$\psi_i(\cdot) = \sum_{j \in \mathcal{S}_i} w_{ij} \phi(\| \cdot - \vec{x}_j \|) = \sum_{k=1}^{\sigma} w_{i,\mathcal{S}_i(k)} \phi(\| \cdot - \vec{x}_{S_i(k)} \|), \tag{6}$$

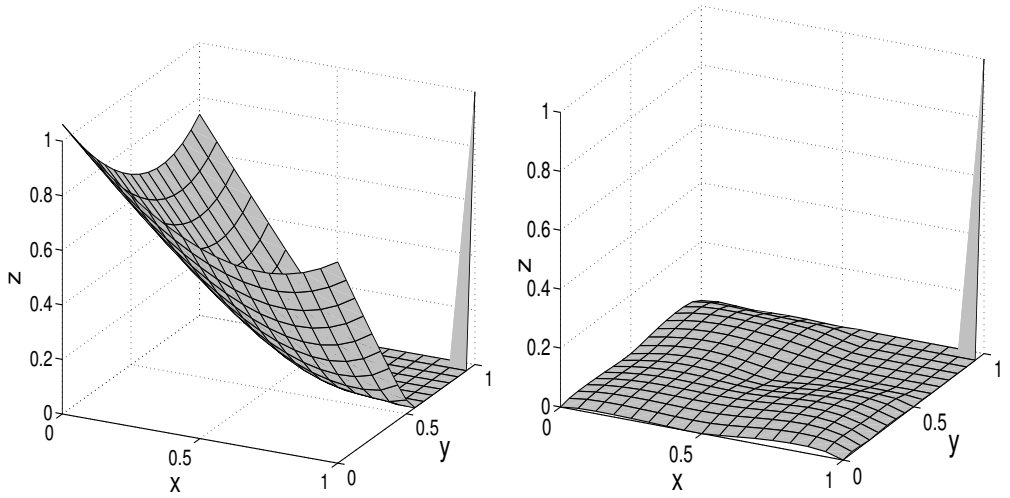

Fig. 1. On the left we have the ACBF based on a pure local strategy and on the right we have the ACBF based on local centers and special points.

satisfies the *cardinal condition*: $\psi_i(\vec{x}_i) = 1$, and $\psi_i(\vec{x}_j) = 0$ for $i \neq j \in \mathcal{S}_i$. The other elements w_{ij} are set to zero for $j \notin \mathcal{S}_i$.

Generally speaking, the index set $\mathcal{S}_i$ should be chosen from the local centers and some special points for both the interpolation [3] and RBF-PDE problem [31]. Choosing $\mathcal{S}_i$ to be the index set corresponding to local points only, yields poor results far away from $\vec{x}_i$. However, the addition of a widely scattered set of special points within the domain counter-acts the growth of the ACBF; see Figure 1.

Each row of the preconditioner W has only σ non-zero entries. These are computed by solving the system

$$B_i^T \vec{w}_i = \vec{e}_i, \tag{7}$$

where $\vec{e}_i$ is the i-th standard basis vector of length σ and

$$B_i = \begin{bmatrix} A_{s_i(1),s_i(1)} & A_{s_i(1),s_i(2)} & \cdots & A_{s_i(1),s_i(\sigma)} \\ A_{s_i(2),s_i(1)} & A_{s_i(2),s_i(2)} & \cdots & A_{s_i(2),s_i(\sigma)} \\ \vdots & \vdots & & \vdots \\ A_{s_i(\sigma),s_i(1)} & A_{s_i(\sigma),s_i(2)} & \cdots & A_{s_i(\sigma),s_i(\sigma)} \end{bmatrix} \in \mathbb{R}^{\sigma \times \sigma}. \tag{8}$$

The i-th row of W is then given by

$$W_{ij} = \begin{cases} w_{i,k} \text{ if } j = s_i(k) \text{ for } k = 1, \ldots, \sigma, \\ 0 \quad \text{otherwise.} \end{cases}$$

Beatson, Cherrie, and Mouat [3] cast the RBF interpolation problem in terms of a set of ACBFs. For each center and corresponding index set, the cardinal condition (6) is then enforced on the same set. This results in a $\sigma \times \sigma$ matrix system. The set up cost is $\mathcal{O}(\sigma^3 N)$ when the direct method is used to solve the N $\sigma \times \sigma$ systems. We apply this idea to RBF-PDE matrix and name the resulting scheme *minimal-ACBF*.

2.2 Approximate cardinal basis functions based on decay elements, DE-ACBF

In the same paper, the far-field expansion of RBFs is also used to construct an ACBF that satisfies a $|\vec{x}|^{-3}$ decay condition. In two dimensions, let $\vec{x} = (x, y)$ and $\vec{x}_i = (\xi_i, \eta_i)$. Consider the center $\vec{x}_i$ as fixed, we choose w_{ij} in such a way that all the lower order terms in the far field expansion vanish for all $\vec{x} = (x, y)$. This results in the *decay element* ACBF (DE-ACBF). Using the truncated far field expansion of a single multiquadric basis function (see [16]) centered at $\vec{x}_i$,

$$\begin{aligned}
&\phi_i(\vec{x}) = \phi(\|\vec{x} - \vec{x}_i\|) = \sqrt{(x-\xi_i)^2 + (y-\eta_i)^2 + c^2} \\
&= \sqrt{x^2+y^2} - \frac{\eta_i y + \xi_i x}{\sqrt{x^2+y^2}} + \frac{1}{2}\frac{(\eta_i^2 + c^2)x^2 + (\xi_i^2 + c^2)y^2 - 2\xi_i\eta_i xy}{(x^2+y^2)^{\frac{3}{2}}} \\
&\quad + \frac{1}{2}\frac{(\xi_i x + \eta_i y)\{(\eta_i^2 + c^2)x^2 + (\xi_i^2 + c^2)y^2 - 2\xi_i\eta_i xy\}}{(x^2+y^2)^{\frac{5}{2}}} + \mathcal{O}(\|\vec{x}\|^{-3}),
\end{aligned} \tag{9}$$

the ACBF expression (6) becomes

$$\begin{aligned}
&\psi_j(\vec{x}) = \sum_{j\in\mathcal{S}_i} w_{ij}\sqrt{x^2+y^2} - \sum_{j\in\mathcal{S}_i} w_{ij}\frac{\eta_i y + \xi_i x}{\sqrt{x^2+y^2}} \\
&+ \frac{1}{2}\sum_{j\in\mathcal{S}_i} w_{ij}\frac{(\eta_i^2 + c^2)x^2 + (\xi_i^2 + c^2)y^2 - 2\xi_i\eta_i xy}{(x^2+y^2)^{\frac{3}{2}}} \\
&+ \frac{1}{2}\sum_{j\in\mathcal{S}_i} w_{ij}\frac{(\xi_i x + \eta_i y)\{(\eta_i^2 + c^2)x^2 + (\xi_i^2 + c^2)y^2 - 2\xi_i\eta_i xy\}}{(x^2+y^2)^{\frac{5}{2}}} + \mathcal{O}(\|\vec{x}\|^{-3}).
\end{aligned} \tag{10}$$

The resulting decay property follows from applying the ten following constraints on the coefficients of W

$$\sum_{j\in\mathcal{S}_i} w_{ij}\vec{x}_j^{\gamma} = 0,$$

for all $\gamma = (\gamma_1, \gamma_2) \in \mathbb{N}_0^2$ such that $\gamma_1 + \gamma_2 \leq 3$.

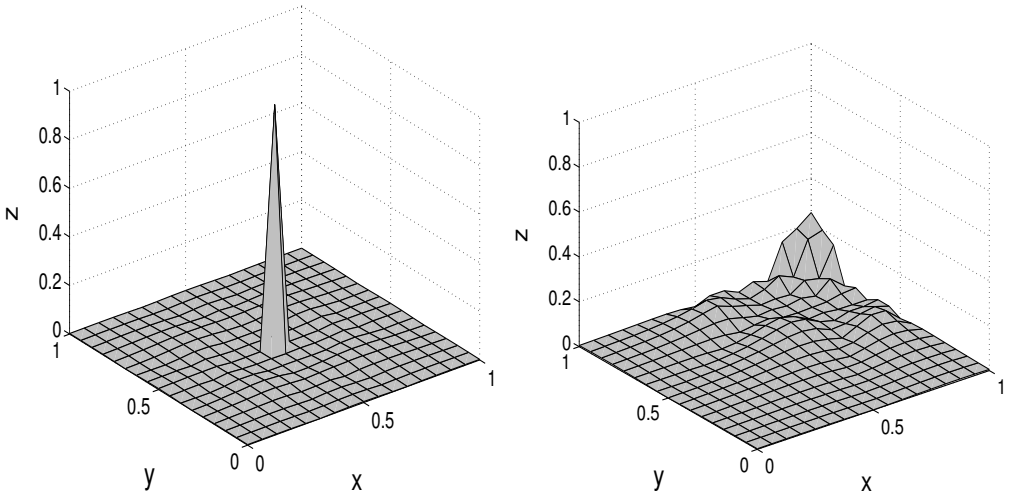

Fig. 2. On the left we have the DE-ACBF with a symmetric node set, on the right we have the DE-ACBF with a non-symmetric node set.

This approximation proves to be effective when the node set is approximately symmetric. At points where a symmetric node set cannot be found (i.e., points near the boundary) decay proves to be much slower; see Figure 2. Such bad basis elements are then discarded and replaced with the ACBF computed in Section 2.1. In order to verify the decay rate of a given ACBF we look at

$$T = \sum_{j \in \mathcal{S}_i} |\psi_j(\vec{x}_i) - \delta_{ij}|.$$

This value will be small if the ACBF ψ_j is close to δ and thus we test $T < \mu$ for some μ usually taken to be 0.5. If the function fails this test then we use an ACBF based on both local and special points.

To extend the discussion from the interpolation problem to PDEs, we change our spline space from $\{\phi_i(\cdot)\}_{i=1}^N$ to

$$\{\Psi_i(\cdot)\}_{i=1}^N = \{\mathcal{L}\phi_i(\cdot)\}_{i=1}^{N_I} \cup \{\mathcal{B}\phi_i(\cdot)\}_{i=1}^{N_B}. \tag{11}$$

The DE-ACBF method can be extended to PDEs, as was introduced by Mouat [35]. He decoupled the interior nodes from the boundary nodes as the same far-field expansion does not apply to both differential operators.

For interpolation problems, we require the far field expansion of the radial basis function, ϕ, being used. When solving PDEs, we also need the far field expansion of $\mathcal{L}\phi$ and $\mathcal{B}\phi$. For the example of Poisson equation in Section 4, we require the far field expansion of the Laplacian of the multiquadric basis function,

$$\nabla^2\phi_i(\vec{x}) = ((x-\xi_i)^2 + (y-\eta_i)^2 + 2c^2)((x-\xi_i)^2 + (y-\eta_i)^2 + c^2)^{-\frac{3}{2}}.$$

This is a generalised multiquadric, see Cherrie, Beatson, and Newsam [16], whose far field expansion is

$$\nabla^2\phi_i(\vec{x}) = \frac{1}{\sqrt{x^2+y^2}} + 3\frac{\eta_i y + \xi_i x}{\sqrt{x^2+y^2}} - 2\frac{\eta_i y + \xi_i x}{(x^2+y^2)^{\frac{3}{2}}} + \mathcal{O}(\|\vec{x}\|^{-3}).$$

The DE-ACBF expression then becomes

$$\begin{aligned}\psi_j(\vec{x}) = &\sum_{j\in\mathcal{S}_i} w_{ij}\frac{1}{\sqrt{x^2+y^2}} + 3\sum_{j\in\ Si} w_{ij}\frac{\eta_i y + \xi_i x}{\sqrt{x^2+y^2}} \\ &-2\sum_{j\in\mathcal{S}_i} w_{ij}\frac{\eta_i y + \xi_i x}{(x^2+y^2)^{\frac{3}{2}}} + \mathcal{O}(\|\vec{x}\|^{-3}).\end{aligned}$$

Thus we enforce the following 3 constraints on the coefficients of W

$$\sum_{j\in\mathcal{S}_i} w_{ij}\vec{x}_j^{\gamma} = 0,$$

for all $\gamma = (\gamma_1, \gamma_2) \in \mathbb{N}_0^2$ such that $\gamma_1 + \gamma_2 \leq 1$.

While the construction of other preconditioners has no dependence on the basis function used or the problem being solved, construction of a preconditioner using DE-ACBF's depends upon both the basis function and the problem being solved. Similar results are also available for the thin-plate spline, see [4,5,7,35].

2.3 Approximate cardinal basis functions based on solving least squares problems, LS-ACBF

Ling and Kansa [31] proposed a simple preconditioning scheme that is based upon constructing the least-squares approximate cardinal basis functions (LS-ABCF) that targets ill-conditioned problems. The cardinal condition (6) is enforced on the whole data set that results in N least-squares problems,

$$B_i^T\vec{w}_i = \vec{e}_i, \tag{12}$$

where $\vec{e}_i$ is the i-th standard basis vector and

$$B_i = \begin{bmatrix} A_{s_i(1),1} & A_{s_i(1),2} & \dots\dots\dots & A_{s_i(1),N} \\ A_{s_i(2),1} & A_{s_i(2),2} & \dots\dots\dots & A_{s_i(2),N} \\ \vdots & \vdots & & \vdots \\ A_{s_i(\sigma),1} & A_{s_i(\sigma),2} & \dots\dots\dots & A_{s_i(\sigma),N} \end{bmatrix} \in \mathbb{R}^{\sigma \times N}. \tag{13}$$

is a submatrix of A, and

$$\vec{w}_i = [w_{i,1}, \dots, w_{i,\sigma}]^T,$$

are the σ non-zeros elements of the i-th row of the preconditioner W. The rows of W are then given by

$$W_{ij} = \begin{cases} w_{i,k} \text{ if } j = s_i(k) \text{ for } k = 1, \dots, \sigma, \\ 0 \quad \text{otherwise.} \end{cases} \tag{14}$$

The method is practical since it does not require any a priori expansion of the RBF being used, and is not limited to just PDEs. The least squares problems are solved with either the normal equation, QR factorization with updates, or SVD factorization with updates. For instance, a considerable amount of storage and computational savings is obtained by the observation that only $\mathcal{O}(\sigma N)$ elements of AA^T are needed for all N normal equations when the normal equation approach is employed. The total cost of constructing the preconditioner is $\mathcal{O}(\sigma N^2 + \sigma^3 N)$ where the first term is the cost of finding all elements of the N normal equations, and the second term is that of solving the N normal equations using direct method. The cost of solving a Poisson's equation using preconditioned GMRES is seen numerically to be $\mathcal{O}(N^{2.11})$ flops in the same paper. Furthermore, in [32] the LS-ACBF preconditioning scheme is coupled with domain decomposition method (DDM) using a customized updating technique developed by the same authors.

2.4 Approximate cardinal basis functions as right-hand preconditioner, 2 sided-ACBF

The RBF-PDE preconditioning scheme found in Section 2.3 can be generalized as a class of preconditioners. We impose the cardinal condition (6) on a set $\mathcal{G}_i$ where $\mathcal{S}_i \subseteq \mathcal{G}_i \subseteq X$ giving

$$\psi_i(\vec{x}_i) = 1, \text{ and } \psi_i(\vec{x}_j) = 0 \text{ for } j \in \mathcal{G}_i,\ j \neq i$$

for some index subset $\mathcal{G}_i = [g_i(1), \ldots, g_i(\gamma)]$.

For the center $\vec{x}_i$, the ACBF expression (6) is determined by the choice of the associated index set $\mathcal{S}_i$. An ACBF preconditioner is now determined by the choice of $\mathcal{G}_i$ on which we impose, in the least squares sense, the cardinal condition.

When $\mathcal{G}_i = \mathcal{S}_i$, it is the *minimal-ACBF* in Section 2.1; when $\mathcal{G}_i = X$, we have the LS-ACBF found in Section 2.3. When $\mathcal{S}_i \subset \mathcal{G}_i \subset X$, there are many possible choices. Note that there are more equations than the number of unknowns and N least-squares problems need to be solved. The least-squares problem associated with the center x_i is

$$\tilde{B}_i^T \vec{w}_i = \vec{e}_i, \tag{15}$$

where

$$\tilde{B}_i = \begin{bmatrix} A_{s_i(1),g_i(1)} & A_{s_i(1),g_i(2)} & \cdots & A_{s_i(1),g_i(\gamma)} \\ A_{s_i(2),g_i(1)} & A_{s_i(2),g_i(2)} & \cdots & A_{s_i(2),g_i(\gamma)} \\ \vdots & \vdots & & \vdots \\ A_{s_i(\sigma),g_i(1)} & A_{s_i(\sigma),g_i(2)} & \cdots & A_{s_i(\sigma),g_i(\gamma)} \end{bmatrix} \in \mathbb{R}^{\sigma \times \gamma}. \tag{16}$$

Although $\tilde{B}$ is still a submatrix of A, the normal matrix $\tilde{B}\tilde{B}^T$ is no longer a submatrix of AA^T. Therefore, the data we obtain at one center cannot be reused in the other centers.

The idea of preconditioning from both sides using LS-ACBF methods was first introduced in [31]. However, the method is concluded to be not practical due to the high set up cost. The ultimate goal is to further improve the condition of the RBF-PDE system while keeping the set up cost low. Since the cost of constructing the minimal-ACBF (i.e. $\mathcal{G}_i = \mathcal{S}_i$) is relatively low in comparison of LS-ACBF, one could re-apply the idea of minimal-ACBF to get a right hand preconditioner. Once the left-hand preconditioner W is found, one can apply a similar process to the columns of WA to construct a right-hand preconditioner $V \sim (WA)^{-1}$. We define C_i to be the submatrix of WA formed by the $\mathcal{S}_i$-th columns and $\mathcal{S}_i$-th rows of WA. Since the same subset $\mathcal{S}_i$ used in the construction of W is re-used for every center, we avoid extra computation and

storage for the index sets. For $C_i \in \mathbb{R}^{\sigma\times\sigma}$ given by

$$C_i = \begin{bmatrix} WA_{s_i(1),s_i(1)} & WA_{s_i(1),s_i(2)} & \dots & WA_{s_i(1),s_i(\sigma)} \\ WA_{s_i(2),s_i(1)} & WA_{s_i(2),s_i(2)} & \dots & WA_{s_i(2),s_i(\sigma)} \\ \vdots & \vdots & \ddots & \vdots \\ WA_{s_i(\sigma),s_i(1)} & WA_{s_i(\sigma),s_i(2)} & \dots & WA_{s_i(\sigma),s_i(\sigma)} \end{bmatrix}, \tag{17}$$

the corresponding square system is

$$C_i\, \vec{v}_i = \vec{e}_i,$$

where $\vec{v}_i = [v_{i,1}, v_{i,2}, \dots, v_{i,m}]$ are the non-zero elements of the i-th column of V. The columns of V are then given by

$$V_{ij} = \begin{cases} v_{j,k} \text{ if } i = s_j(k) \text{ for } k = 1, \dots, \sigma, \\ 0 \quad \text{otherwise.} \end{cases}$$

Not all the elements of WA are required to solve all N systems. According to the observation Ling and Kansa made in [31], the cost of finding all necessary elements of WA would by $\mathcal{O}(\sigma^2 N)$. The total cost of solving N systems of equations of the size $\sigma \times \sigma$ would be $\mathcal{O}(\sigma^3 N)$ if a direct method is employed. Similar to the left-hand preconditioner W, the preconditioner V only has σ nonzero elements in each column; the matrix-vector product of V can be performed in $\mathcal{O}(\sigma N)$ flops and an $\mathcal{O}(\sigma N)$ storage.

3 Fast evaluation

An important work of Beatson and coworkers [2,16] is the fast RBF matrix-vector product algorithm. The evaluation of the MQ expansion at any $\vec{x} \in \mathbb{R}^d$ costs $\mathcal{O}(N \log N)$ flops; this is much faster than the $\mathcal{O}(N^2)$ flops for direct matrix-vector multiplications. This algorithm is also applicable for the RBF-PDE matrix A and its transpose A^T. Fast multiplication algorithms can be used to reduce the cost of each GMRES iteration. However, a customized algorithm is *not* necessary for different PDEs as it commonly is for traditional finite difference method (FDM), finite element method (FEM), and finite volume method (FVM) applications. Once the algorithms are constructed for all the involved derivatives individually, they could be reused for other PDEs of the same order.

Fast evaluation can easily be applied to PDE problems through the computation of the far field expansion of the differential operator acting on a radial basis function. In order to explain the process involved we look at an interpolation problem $\Phi\, d = f$ on some data $\{x_i\}_{i=1}^N \subset \mathbb{R}^d$ and a chosen radial basis function, in this case the multiquadric. The resulting solution allows us to evaluate the approximation to f at any point x by computing

$$f(x) \approx \sum_{i=1}^{N} \alpha_i \sqrt{||x - x_i||^2 + c^2}.$$

However, this requires a huge amount of work for large N. This section aims to reduce this work by explicitly evaluating only at those points nearby, the near field, and approximating the effects of far away points, the far field. This method was introduced by Greengard and Rokhlin [20] in the case of potentials and was documented by Beatson and Newsam [2] when studying the thin plate spline, and further by Beatson, Cherrie and Newsam [16] when studying the multiquadric.

For the remainder of this section we work in two dimensions or more precisely we work on the complex plane, and we discuss only the multiquadric. We call a series "Laurent-like" if it valid at all points outside a particular disc.

The method for evaluating the far field requires us to decompose the domain uniformly retaining information at all levels. We assume the data is within the unit square $[0, 1] \times [0, 1]$ and split the region into four equal squares. These squares will henceforth be known as panels, in particular the first level panels. We then split each of these panels into four equal squares known as the second level panels. This method is applied recursively until the mth level is reached where m can be chosen by the user typically depending on the amount of data N. This structure, in two dimensions, is known as a quadtree.

We can now define which panels are associated with the near field and which are associated with the far field.

Definition 1 *A point x is said to be far away from a panel T if they are separated by at least one panel at the same level as T.*

Definition 2 *A panel Q is said to be on the evaluation list of a panel T if*

(1) Q is on the same or higher level than T,
(2) every point in T is far away from Q,
(3) T contains a point which is not far away from the parent of Q.

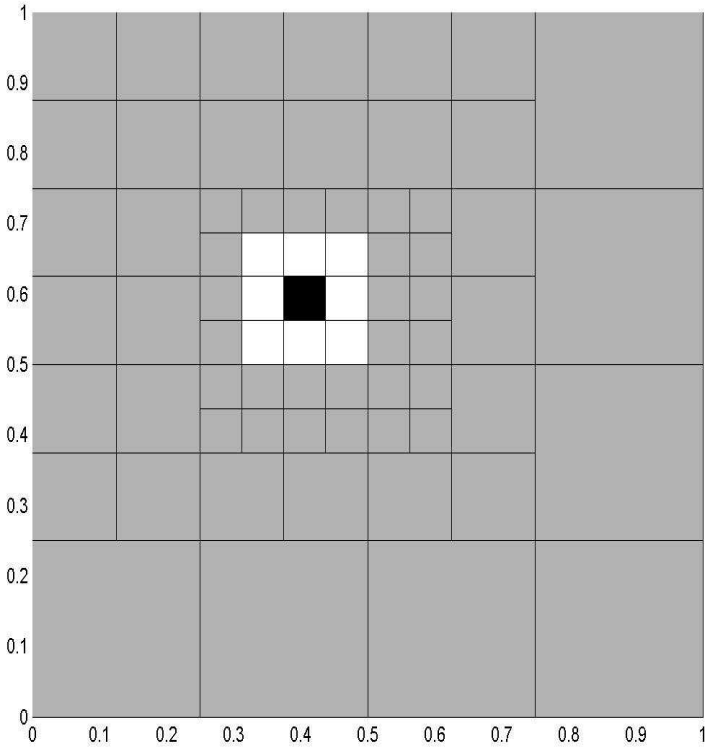

Fig. 3. The black square is the panel T and the grey squares are those panels on the evaluation list of T. Therefore the grey squares are associated with the far field and the black and white squares are associated with the near field.

We associate with each panel Q a function s_Q defined by

$$s_Q(x) = \sum_{i:x_i \in Q} d_i \sqrt{||x - x_i||^2 + c^2}.$$

The far field, s_f, is now defined as the sum of s_Q such that Q is on the evaluation list of the panel T containing x. The near field, s_n is the remainder, i.e. $s_n = s - s_f$. Figure 3 illustrates which panels are associated with the near and far field. We now intend to approximate s_Q as a Laurent-like expansion about the center of Q. If this is done for all panels on the evaluation list of some panel T, then we can simply sum the series to compute the far field. We now present a lemma giving the Laurent-like expansion of a single multiquadric.

Lemma 3 *Let $t \in \mathbb{C}$ and $c \geq 0$ and*

$$\Phi_t(z) = \sqrt{|z - t|^2 + c^2}.$$

Then for all $z \in \mathbb{C}$ *with* $|z| > \sqrt{|t|^2 + c^2}$

$$\Phi_t(z) = \sum_{l=0}^{\infty} \frac{P_l(|t|^2 + c^2, -2\Re(t\overline{z}), |z|^2)}{|z|^{2l-1}},$$

where

$$P_l(a, b, c) = \sum_{j=\lfloor \frac{l+1}{2} \rfloor}^{l} \begin{pmatrix} \frac{1}{2} \\ j \end{pmatrix} \begin{pmatrix} j \\ l-j \end{pmatrix} b^{2j-l}(ac)^{l-j}.$$

Furthermore, for any $p \geq 0$

$$\left| \Phi_t(z) - \sum_{l=0}^{p+1} \frac{P_l(|t|^2 + c^2, -2\Re(t\overline{z}), |z|^2)}{|z|^{2l-1}} \right| \leq 2\sqrt{|t|^2 + c^2} \left(\frac{\sqrt{|t|^2 + c^2}}{|z|} \right)^{p+1} \frac{|z|}{|z| - \sqrt{|t|^2 + c^2}}.$$

A proof of this can be found in Beatson, Cherrie and Newsam [16]. The next lemma is merely an extension of Lemma 3 and requires no proof.

Lemma 4 *Let* $t_i \in \mathbb{C}$, $|t_i| \leq r$ *and* $d_i \in \mathbb{R}$ *for* $1 \leq i \leq N$, $c \geq 0$ *and* s *be the multiquadric*

$$s(z) = \sum_{i=1}^{N} d_i \sqrt{|z - t_i|^2 + c^2}.$$

We define the polynomial Q_l *to be*

$$Q_l(z) = \sum_{i=0}^{N} d_i P_l(|t_i|^2 + c^2, -2\Re(t_i\overline{z}), |z|^2),$$

where P_l *is the polynomial defined in Lemma 3. Let* $p \in \mathbb{N}_0$ *and*

$$s_p(z) = \sum_{l=0}^{p+1} \frac{Q_l(z)}{|z|^{2l-1}}.$$

Then, for all z such that $|z| > \sqrt{r^2 + c^2}$,

$$|s(z) - s_p(z)| \leq 2M\sqrt{r^2 + c^2} \left(\frac{\sqrt{r^2 + c^2}}{|z|} \right)^{p+1} \frac{|z|}{|z| - \sqrt{r^2 + c^2}},$$

where $M = \sum_{i=0}^{N} |d_i|$.

At this point we can evaluate the Laurent-like expansion of s_Q centered at the origin. By making the substitution $z = x - u$ in Lemma 3 where u is the center of the panel about which the expansion is taken, we find that the terms in the series involve convolutions as discussed in [16, Section 7]. An alternative approach is to evaluate the series about the center in the first instance. Therefore we shift the center of a given panel to the origin and all points in that panel will move close to the origin. Computing the series centered at the origin using Lemma 4 is thus equivalent to computing the series about the center of the panel. This has to be done for every panel at all levels in the domain.

At this point we have an expansion for all panels at all levels centered at the center of the panel. We are now ready for evaluation.

The error in the truncated laurent like series expansion has been shown to be $\mathcal{O}(\sigma^p)$ for some $\sigma < 1$. We therefore choose $p \approx \frac{\log \epsilon}{\log \sigma}$ hence the error is $\mathcal{O}(\epsilon)$. It is suggested that $\sigma = \frac{\sqrt{2}}{3}$; see Greengard and Rokhlin [20]. Assuming the distribution of points is approximately uniform, the number of points in the near field is $\mathcal{O}(1 + N4^{-m})$. Therefore to make this $\mathcal{O}(1)$ we set $m \approx \log_4 N$.

Set-up of this hierarchical scheme should proceed according to the following algorithm.

Algorithm 1 Setup

(1) Set m, the number of levels of the quadtree, to be the nearest integer to $\log_4 N$ and choose p, the number of terms, to be the nearest integer to $\frac{\log \epsilon}{\log \frac{\sqrt{2}}{3}}$ where ϵ is the desired accuracy,
(2) form the quadtree down to m levels,
(3) compute the approximation s_Q about the origin for all the lowest level panels,
(4) shift the center of expansion from the origin to the center of Q for all the lowest level panels,
(5) form the expansion for the higher level panels.

We then need to compute the interpolant at some point x using the following

algorithm.

Algorithm 2 Evaluation

(1) Locate the panel T containing x,
(2) find the evaluation list E of T,
(3) evaluate $s(x)$ by summing the far field and the near field, i.e.

$$s(x) = \sum_{Q \in E} s_Q + \sum_{i: x_i \in T \cup W} \sqrt{||x - x_i||^2 + c^2}$$

where W is the set of fine level panels neighboring T.

4 Numerical examples

4.1 Steady state problems

We set up the Dirichlet problems of the Poisson, Helmholtz, and Modified Helmholtz equations on the unit square $\Omega = [0, 1]^2$.

$$\nabla^2 u(x, y) + \lambda \cdot u(x, y) = f(x, y), \qquad (x, y) \in \Omega, \tag{18}$$

that has an exact solution

$$u(x, y) = \sin \frac{\pi x}{6} \sin \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4}. \tag{19}$$

For all wave numbers $\lambda \in \mathbb{R}$, the function $f(x, y)$ is obtained by substituting (19) into (18). The Poisson equation ($\lambda = 0$) is studied as in [31] with various shape parameters on both regular and irregular centers placements. When $\lambda > 0$, it is a Helmholtz equation; when $\lambda < 0$, it is a modified Helmholtz equation. The problems are more ill conditioned as $|\lambda|$ increases. We experimented with $\lambda = -100, \ -10, \ 0, \ 100$, and 1000.

We computed the approximate solutions to (18) using the multiquadric RBF with a constant shape parameter $c = \frac{1}{\sqrt{N}}$, on an $N = n \times n$ uniform grid where $n = 17, \ 25, \ 33, \ldots, \ 65$. We used 50 local points plus 9 special points and computed the LS-ACBF, minimal-ABCF, and 2 sided-ACBF preconditioners for all N. The resulting linear systems are solved with GMRES. For all tested values of N and λ, GMRES converges and the number of iterations required for convergence are reported graphically in Figure 4 to Figure 8. Furthermore,

the DE-ACBF is also used on the Poisson problem for comparison in Figure 4.

We remind the reader that the condition number of an unsymmetric matrix is not as informative as in the symmetric case, A small condition number does not necessarily means fast GMRES convergence since convergence depends of the clustering of eigenvalues. However, a good preconditioning scheme must be able to reduce condition number and we included some information here for completeness. Figure 9 reports the condition number of the Poisson problem: the preconditioned matrices with LS-ACBF have the smaller condition number than the 2 sided-ACBF, and lastly the minimal-ACBF. This behavior can also be found in cases of $\lambda = -100,\ -10$, and 1000. Also for these cases, the trend is monotonic like the Poisson equation and the minimal ACBF condition number at $N = 4225$ is about 10 times larger than with LS-ACBF condition number.

The special case is the Helmholtz problem when $\lambda = 100$. From Figure 10, we see that the condition numbers of the original RBF-PDE matrices do not show the monotonic trend with N. Note that LS-ACBF has relatively stable performance in this problem in terms of the reduction of condition numbers. On the other hand, the 2 sided-ACBF fails to reduce condition number when $N = 4225$ implying the 2 sided-ACBF scheme could be unstable.

We observe that LS-ACBF is more efficient when $\lambda = 100$ and $\lambda = -1000$, see Figure 6 and Figure 8, respectively. The LS-ACBF preconditoner in [31] is designed to handle ill conditioned problems; our numerical results further confirm this claim. On the other hand, the minimal-ABCF, and 2 sided-ACBF schemes are more efficient for the other values $\lambda = -10,\ 0$ and 100; small $|\lambda|$ values are associated with relatively better conditioned cases. In most cases, the 2 sided-ACBF scheme converges no faster than the minimal-ABCF scheme *except* the Helmholtz problem with $\lambda = 100$; see Figure 6.

The error was computed on a uniform 81×81 grid. We computed the mean square residual error (MSR) given by $\text{MSR} = \frac{1}{81}\sqrt{\sum_i (U(z_i) - u(z_i))^2}$, and the maximum error (MAX) given by $\text{MAX} = \max_i |U(z_i) - u(z_i)|$, where U is the approximate solution and z_i are the points on an 81×81 grid. The results for all preconditioning schemes are consistent to at least 4 significant figures.

4.2 Time dependent problems

We now consider a diffusion equation on $\Omega = [0, 1]^2$

$$\frac{1}{k}\nabla^2 u(x, y, t) = \frac{\partial}{\partial t} u(x, y, t), \qquad (x, y) \in \Omega, \tag{20}$$

$$u(x,y,t) = 0, \qquad (x,y) \in \partial\Omega, \qquad 0 \le t \le T, \tag{21}$$
$$u(x,y,0) = 1, \qquad (x,y) \in \Omega, \tag{22}$$

where k is the inverse of heat conductivity. The exact solution is given in [11]; the same problem is used in [30] as a test problem. Let $U^\ell(x,y) \approx u(x,y,\ell\delta_t)$, we employ the backward difference scheme to approximate the time derivative:

$$\frac{\partial}{\partial t}u(x,y,t) \approx \frac{U^{\ell+1}(x,y) - U^\ell(x,y)}{\delta_t}. \tag{23}$$

For each time step, the differential equation (20) is approximated by

$$\frac{1}{k}\nabla^2 U^{\ell+1}(x,y,t) = \frac{U^{\ell+1}(x,y) - U^\ell(x,y)}{\delta_t},$$

that results in a sequence of modified Helmholtz equations,

$$\nabla^2 U^{\ell+1}(x,y,t) - \frac{k}{\delta_t}U^{\ell+1}(x,y) = -\frac{k}{\delta_t}U^\ell(x,y), \tag{24}$$

for $\ell = 1, 2, \ldots, L$, with boundary condition given by (21). It is well-known that the accuracy of (23) improves as δ_t decreases. However, this results in a large wave number λ in the modified Helmholtz equation. As we can see from the previous example, this causes ill-conditioned matrix problems. For large heat conductivity k, one is forced to use large time step δ_t in order to keep the wave number to a reasonable size. In this example, we use $k = 1000$ and $\delta_t = 0.01$ to compute the solution when $t = 1$; the effective wave number is $\lambda = -10^5$.

To solve the problem, the left-hand side of (24) is expanded using the multiquadric RBF with 33×33 data centers. The shape parameter is the same as the previous example. The value of U^0 is given by the initial condition (22). The resulting matrix systems are solved with GMRES preconditioned by either LS-ACBF, minimal-ACBF, or 2 sided-ACBF. After each time step, the GMRES solution (i.e., MQ coefficients) is transformed back to function values using the fast multipole method that in turn become the right-hand vector of the next matrix system. Furthermore, the solution at $t = \ell\delta_t$ is used as the initial guess for the matrix system defined at $t = (\ell+1)\delta_t$. The numerical solution at $t = 1$ is given in Figure 11, and the number of GMRES iterations required for each time step is reported in Figure 12.

5 Conclusion

We study different approximate cardinal basis function (ACBF) approaches to precondition RBF-PDE systems. The four methods we study are based on solving square problems (minimal-ACBF), on decay elements (DE-ACBF), solving least squares problems (LS-ACBF), and both sided preconditioning (2 sided-ACBF).

The cost of solving using these schemes can be catalogued as

Set up Cost: minimal-ACBF $<$ LS-ACBF $<$ 2 sided-ACBF,
Run Cost: minimal-ACBF $<$ 2 sided-ACBF $<$ LS-ACBF.

The DE-ACBF is difficult to set up and is only applied to the Poisson problem. However, the *best* preconditioning scheme is problem dependent. From our study of various steady state problems, we found that the LS-ACBF is a better choice for ill conditioned problems. On the other hand, the minimal-ACBF and 2 sided-ACBF are more efficient for better conditioned problems. In such cases, GMRES using the 2 sided-ACBF converges no faster than that using minimal-ACBF. However, the set up and run cost of the 2 sided-ACBF is more expensive than the minimal-ACBF.

For time-dependent problems, if one desires a solution of an ill conditioned problem at large t, the "run cost" should be minimized and LS-ACBF is preferred. On the other hand, if the problem size is small or the problem is well conditioned, one may prefer the minimal-ACBF or even a direct method to save the *human* set up cost. We wish to remind the reader that for large scale FDM, FEM, or FVM problems, the preconditioners that are used on parallel computers are often customized for the problem being solved.

References

[1] B. J. C. Baxter. Preconditioned conjugate gradients, radial basis functions, and Toeplitz matrices. *Comput. Math. Appl.*, 43(3-5):305–318, 2002. Radial basis functions and partial differential equations.

[2] R. Beatson and L. Greengard. A short course on fast multipole methods. In *Wavelets, multilevel methods and elliptic PDEs (Leicester, 1996)*, Numer. Math. Sci. Comput., pages 1–37. Oxford Univ. Press, New York, 1997.

[3] R. K. Beatson, J. B. Cherrie, and C. T. Mouat. Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration. *Adv. Comput. Math.*, 11(2-3):253–270, 1999. Radial basis functions and their applications.

[4] R. K. Beatson, J. B. Cherrie, and D. L. Ragozin. Fast evaluation of radial basis functions: methods for four-dimensional polyharmonic splines. *SIAM J. Math. Anal.*, 32(6):1272–1310 (electronic), 2001.

[5] R. K. Beatson and W. A. Light. Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines. *IMA J. Numer. Anal.*, 17(3):343–372, 1997.

[6] R. K. Beatson, W. A. Light, and S. Billings. Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM J. Sci. Comput.*, 22(5):1717–1740 (electronic), 2000.

[7] R. K. Beatson and G. N. Newsam. Fast evaluation of radial basis functions. I. *Comput. Math. Appl.*, 24(12):7–19, 1992. Advances in the theory and applications of radial basis functions.

[8] M. D. Buhmann. Multivariate cardinal interpolation with radial-basis functions. *Constr. Approx.*, 6(3):225–255, 1990.

[9] M. D. Buhmann. Multivariate interpolation in odd-dimensional Euclidean spaces using multiquadrics. *Constr. Approx.*, 6(1):21–34, 1990.

[10] M. D. Buhmann and C. A. Micchelli. Multiquadric interpolation improved. *Comput. Math. Appl.*, 24(12):21–25, 1992. Advances in the theory and applications of radial basis functions.

[11] H. S. Carslaw and J. C. Jaeger. *Conduction of heat in solids*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1988.

[12] W. Chen. Orthonormal rbf wavelet and ridgelet-like series and transforms for high-dimensional problems. *Int. J. Nonlinear Sci. & Numer. Simulation*, 2(2):155–160, 2001.

[13] W. Chen. Symmetric boundary knot method. *Engng. Anal. Bound. Elem.*, 26(6):489–494, 2002.

[14] W. Chen and M. Tanaka. A meshless, integration-free, and boundary-only RBF technique. *Comput. Math. Appl.*, 43(3-5):379–391, 2002.

[15] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito. Exponential convergence and *h-c* multiquadric collocation method for partial differential equations. *Numer. Methods Partial Differential Equations*, 19(5):571–594, 2003.

[16] J. B. Cherrie, R. K. Beatson, and G. N. Newsam. Fast evaluation of radial basis functions: methods for generalized multiquadrics in $\mathbb{R}^n$. *SIAM J. Sci. Comput.*, 23(5):1549–1571 (electronic), 2002.

[17] B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Adv. Comput. Math.*, In press.

[18] C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. Comput. Math.*, 8(4):381–399, 1998.

[19] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comput.*, 93(1):73–82, 1998.

[20] L. Greengard and V. Rokhlin. A fast algoritm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[21] Y. C. Hon. A quasi-radial basis functions method for American options pricing. *Comput. Math. Appl.*, 43(3-5):513–524, 2002. Radial basis functions and partial differential equations.

[22] Y. C. Hon and X. Z. Mao. A multiquadric interpolation method for solving initial value problems. *J. Sci. Comput.*, 12(1):51–55, 1997.

[23] Y. C. Hon and X. Z. Mao. An efficient numerical scheme for Burgers' equation. *Appl. Math. Comput.*, 95(1):37–50, 1998.

[24] Y. C. Hon and X. Z. Mao. A radial basis function method for solving options pricing models. *Financial Engineering*, 8(1):31–49, 1999.

[25] E. J. Kansa. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates. *Comput. Math. Appl.*, 19(8-9):127–145, 1990.

[26] E. J. Kansa. Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.*, 19(8-9):147–161, 1990.

[27] E. J. Kansa and Y. C. Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Comput. Math. Appl.*, 39(7-8):123–137, 2000.

[28] E. Larsson and B. Fornberg. A numerical study of radial basis functions based solution methods for elliptic pdes. *Comput. Math. Appl.*, In press.

[29] J. Li and Y. C. Hon. Domain decomposition for radial basis meshless methods. *Numer. Methods for PDEs*, 20(3):450–462, 2004.

[30] X. Li and C. S. Chen. A mesh free method using hyperinterpolation and fast fourier transform for solving differential equations. *Engng. Anal. Bound. Elem.*, In review.

[31] L. Ling and E. J. Kansa. A least-squares preconditioner for radial basis functions collocation methods. *Adv. Comput. Math.*, In press.

[32] L. Ling and E. J. Kansa. Preconditioning for radial basis functions with domain decomposition methods. *Math. Comput. Modelling*, In press.

[33] W. R. Madych. Miscellaneous error bounds for multiquadric and related interpolators. *Comput. Math. Appl.*, 24(12):121–138, 1992. Advances in the theory and applications of radial basis functions.

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

[35] C. T. Mouat. *Fast Algorithms and Preconditioning Techniques for Fitting Radial Basis Functions*. PhD thesis, Mathematics Department, University of Canterbury, Christchurch, New Zealand, 2001.

[36] M. J. D. Powell. The theory of radial basis function approximation in 1990. In *Advances in numerical analysis, Vol. II (Lancaster, 1990)*, Oxford Sci. Publ., pages 105–210. Oxford Univ. Press, New York, 1992.

[37] R. Schaback. Error estimates and condition numbers for radial basis function interpolation. *Adv. Comput. Math.*, 3(3):251–264, 1995.

[38] R. Schaback. Multivariate interpolation and approximation by translates of a basis function. In *Approximation theory VIII, Vol. 1 (College Station, TX, 1995)*, volume 6 of *Ser. Approx. Decompos.*, pages 491–514. World Sci. Publishing, River Edge, NJ, 1995.

[39] R. Schaback. On the efficiency of interpolation by radial basis functions. In *Surface fitting and multiresolution methods (Chamonix–Mont-Blanc, 1996)*, pages 309–318. Vanderbilt Univ. Press, Nashville, TN, 1997.

[40] Z. M. Wu and R. Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA J. Numer. Anal.*, 13(1):13–27, 1993.

[41] J. Yoon. Spectral approximation orders of radial basis function interpolation on the Sobolev space. *SIAM J. Math. Anal.*, 33(4):946–958 (electronic), 2001.

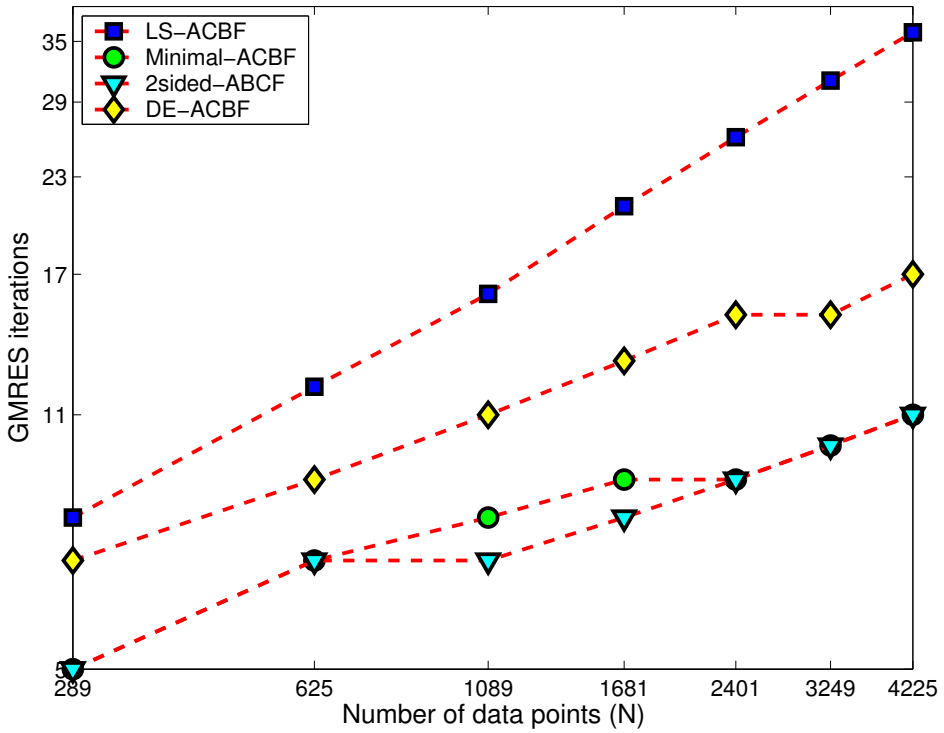

Fig. 4. Number of GMRES iterations required for convergence as a function of N for the Poisson problem $\nabla^2 u = f$.

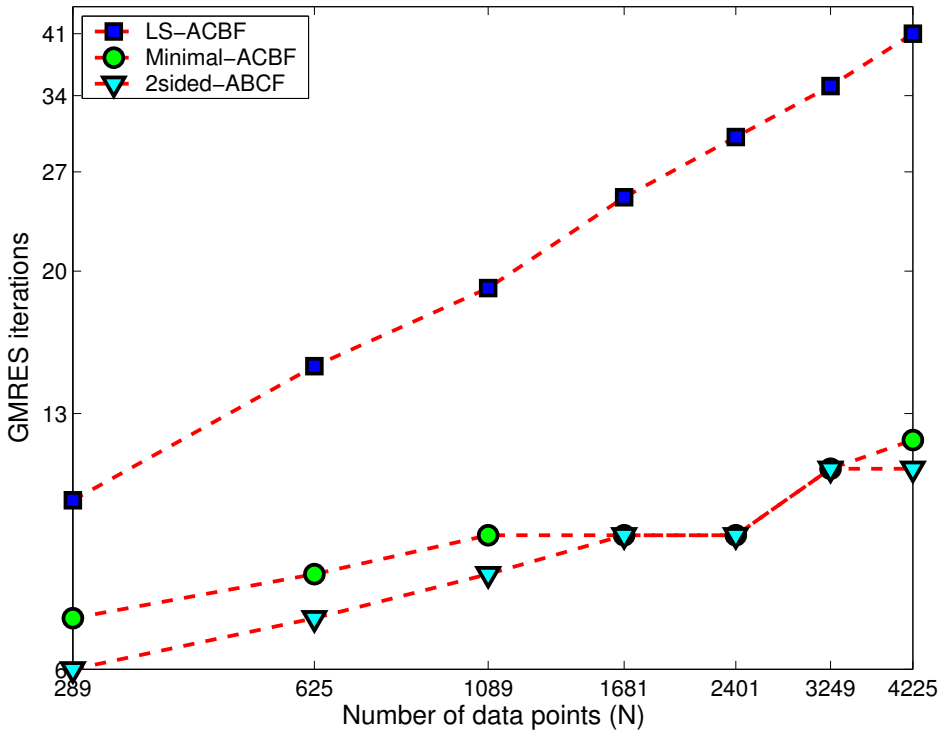

Fig. 5. Number of GMRES iterations required for convergence as a function of N for the Helmholtz problem $\nabla^2 u + 10u = f$.

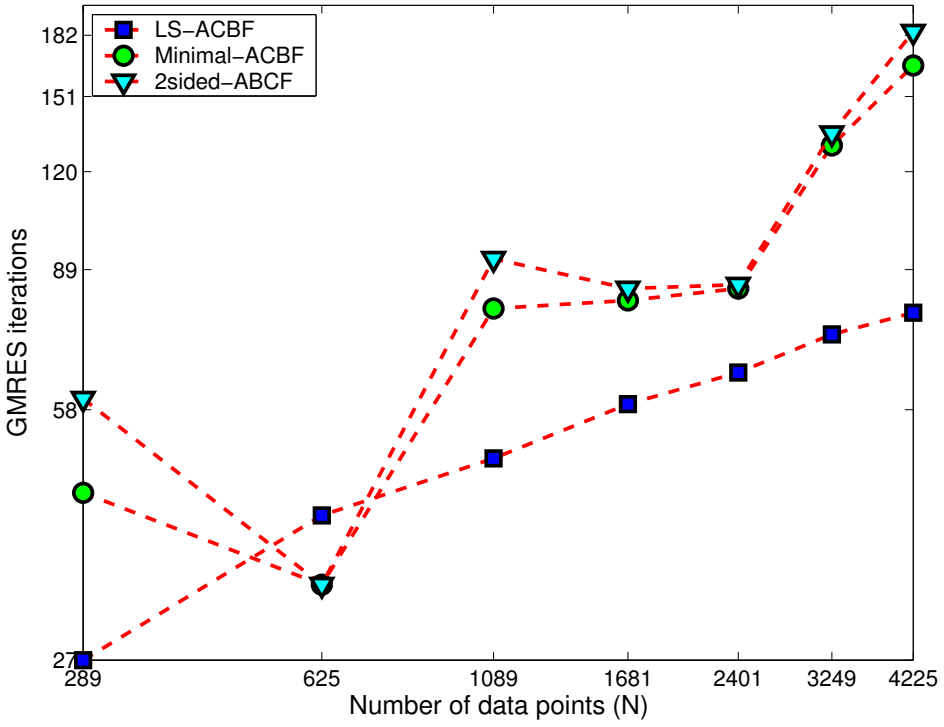

Fig. 6. Number of GMRES iterations required for convergence as a function of N for the Helmholtz problem $\nabla^2 u + 100u = f$.

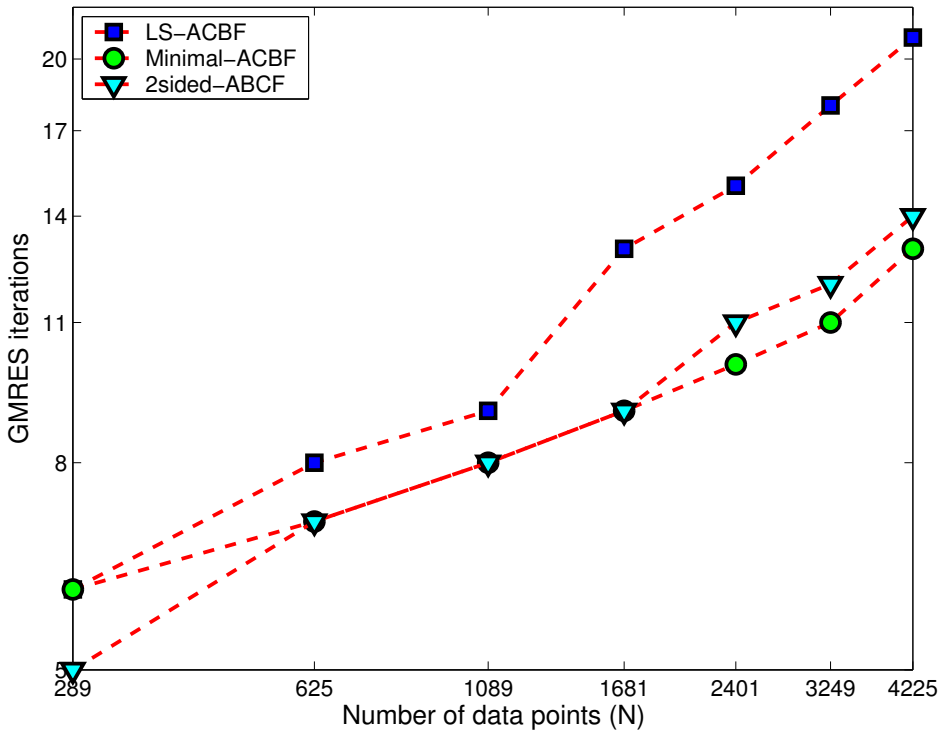

Fig. 7. Number of GMRES iterations required for convergence as a function of N for the modified Helmholtz problem $\nabla^2 u - 100u = f$.

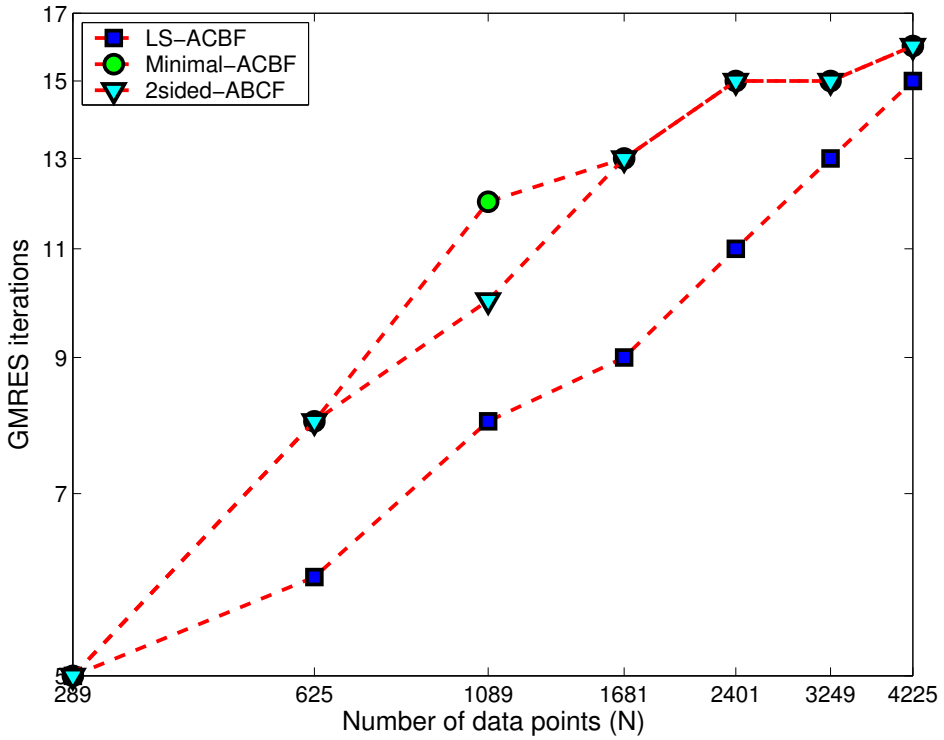

Fig. 8. Number of GMRES iterations required for convergence as a function of N for the modified Helmholtz problem $\nabla^2 u - 1000u = f$.

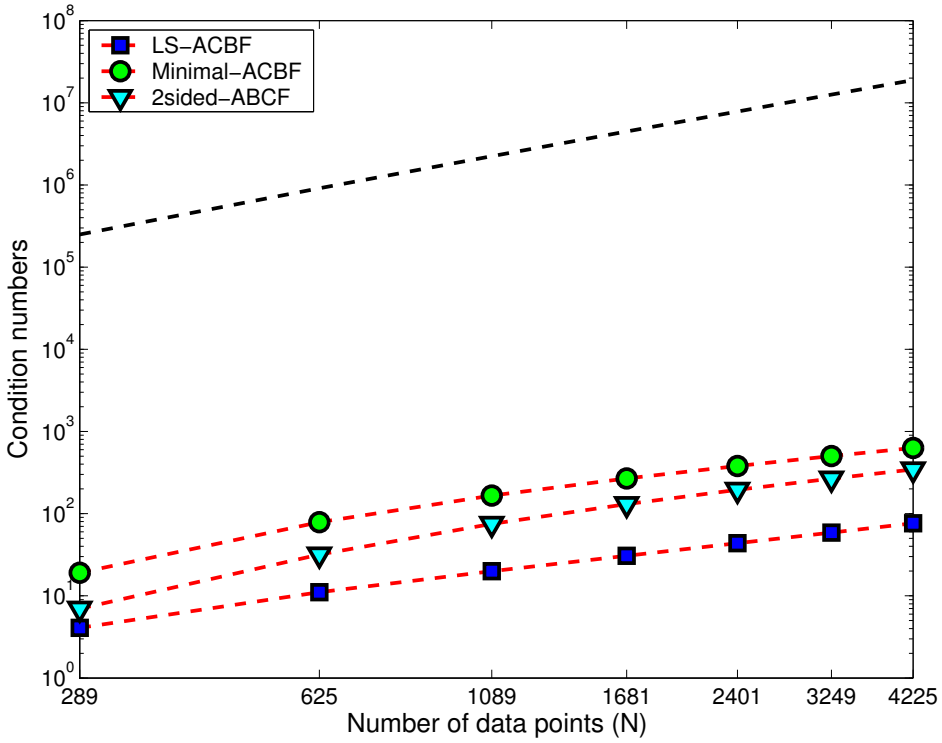

Fig. 9. Condition number of the preconditioned matrices as a function of N for the Poisson problem $\nabla^2 u = f$.

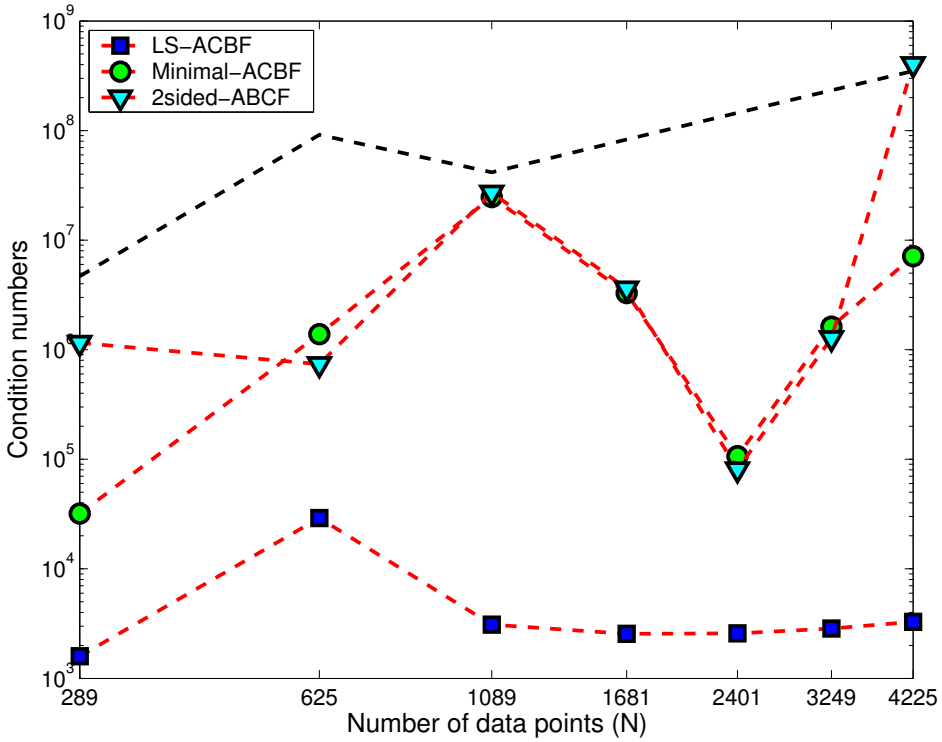

Fig. 10. Condition number of the preconditioned matrices as a function of N for the Helmholtz problem $\nabla^2 u + 100u = f$.

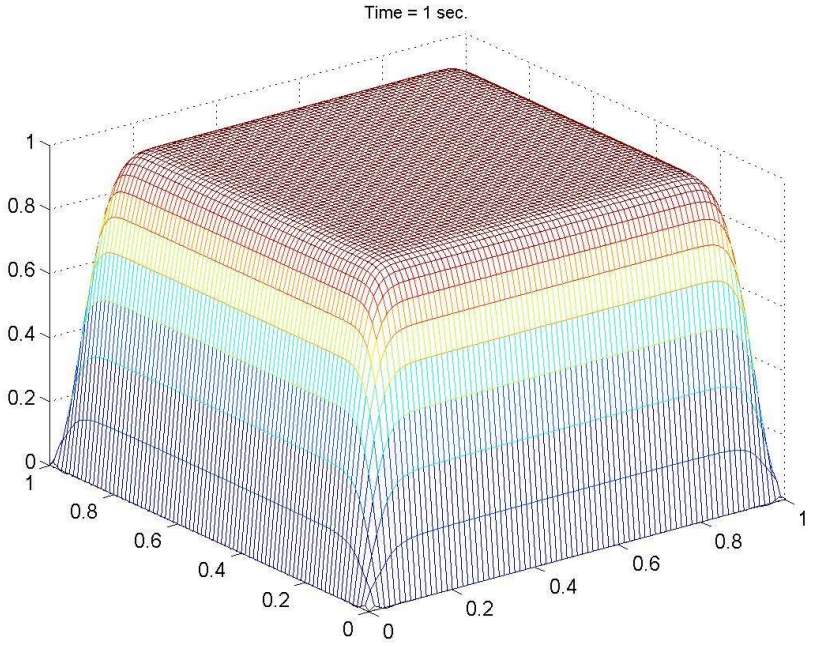

Fig. 11. Numerical solution to the Heat problem (20) with $k = 1000$, and $\delta_t = 0.01$ at $t = 1$ second.

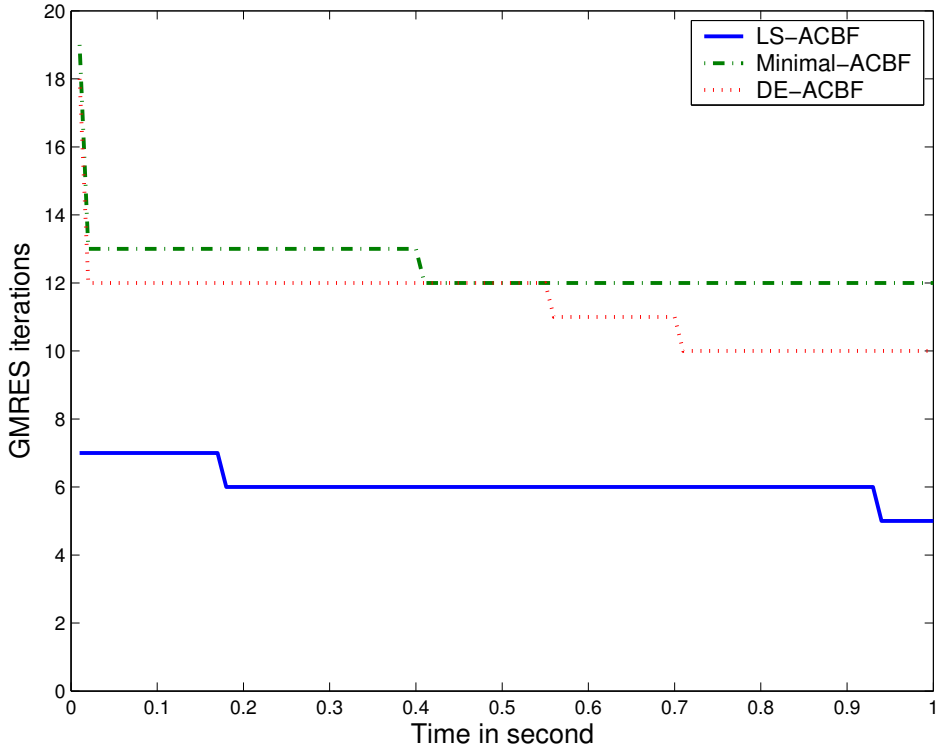

Fig. 12. Number of GMRES iterations required for each time step to find solution in Figure 11.