Multilevel Compact Radial Functions Based Computational Schemes for Some Elliptic Problems

C. S. CHEN
Department of Mathematical Sciences
University of Nevada, Las Vegas
Las Vegas, NV 89154, U.S.A.

M. GANESH
School of Mathematics
University of New South Wales
Sydney, NSW 2052, Australia

M. A. GOLBERG
517 Bianca Bay Street
Las Vegas, NV 89144, U.S.A.

A. H.-D. CHENG
Department of Civil Engineering
University of Mississippi
University, MS 38677, U.S.A.

Abstract—Compactly supported radial basis functions (CS-RBFs) have been recently introduced in the context of the dual reciprocity method as a possible cure of dense matrices and ill-conditioning problems when using the classical radial basis functions. However, the support scaling factor and slow convergence rate of the CS-RBFs have also raised issues on the effectiveness of the CS-RBFs. In this paper, two multilevel schemes have been proposed to alleviate these problems. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords—The method of fundamental solutions, Particular solutions, Compactly supported radial basis functions, Multilevel scheme, Dual reciprocity method, Boundary element method.

1. INTRODUCTION

The idea of solving PDEs numerically based on the radial basis functions (RBFs) was first introduced by Kansa [1] in early the 1990s. In recent years, the application of the RBFs has made its way to the boundary element community. In particular, the dual reciprocity method (DRM) has made significant advances due to the introduction of RBFs. As a result of the rapid development of the DRM, the early version of the basis function $1 + r$ [2] has been largely regarded as obsolete. The rich class of RBFs has been widely used to replace $1 + r$ in the DRM literature due to the improvement in accuracy and solid mathematical foundation.

The authors wish to thank the referees for their helpful comments and suggestions.
The most popular RBFs are multiquadrics, \( \varphi(r) = \sqrt{r^2 + c^2} \), thin plate splines, \( \varphi(r) = r^2 \log r \), and Gaussians, \( \varphi(r) = e^{-c^2r^2} \) which are all globally supported. Although these RBFs exhibit excellent interpolation properties, they also present some numerical drawbacks. It is well known that these globally supported basis functions lead to dense systems which can be highly ill-conditioned, especially for large scale industrial problems. Recent tests also showed that when iterative methods are used to solve the collocation matrices, they failed to converge for some radial basis functions [3]. For RBFs such as multiquadrics and Gaussians, there exists a so-called “principle of uncertainty” [4] which means that good convergence can only be achieved at the expense of instability. Computational efficiency is also a concern due to the dense matrices. In practice, RBFs with global support are recommended for only a few hundred interpolation points. For large scale problems, tens of thousands of interpolation points are often required. Hence, there is a need to employ some type of localization schemes such as domain decomposition [5,6]. In the mid-1990s, locally supported RBFs [4,7–9] first appeared in the approximation theory literature. It turns out that the most popular compactly support RBFs (CS-RBF) are the ones introduced by Wendland [8]. They were regarded as a cure to the problem for the dense and ill-conditioned matrices indicated above. Several researchers recently implemented the newly constructed CS-RBFs for solving various kinds of PDEs numerically [3,10–13]. But several difficulties of the CS-RBFs have been observed:

(i) the accuracy and efficiency depends on the scale of the support and the rule for determining the scale of support is uncertain.
(ii) the convergence rate of CS-RBFs is low.

In order to obtain a sparse matrix system, the support needs to be small; then the interpolation error becomes unacceptable. When the support is large enough to make the error acceptable, the matrix system becomes dense and computational inefficiency is obvious. As a result, the use of the CS-RBFs with a fixed support is not recommended [12]. In [4,12,14], a multilevel scheme was introduced to handle the uncertainty of choosing the size of support (or scaling factor). Fasshauer [12] employed multilevel CS-RBFs in the context of a Hermite collocation method. In his approach, a smoothing scheme was also employed to ensure the satisfactory performance of multilevel CS-RBFs. However, the process of smoothing is not a trivial task.

It is the purpose of this paper to present two multilevel CS-RBFs-based computational approaches in the context of the DRM to solve certain classes of linear inhomogeneous elliptic partial differential equations (PDEs). In the DRM, we use the CS-RBFs to approximate the inhomogeneous term instead of the solution of PDEs as proposed in [12]. As a result, no smoothing at each level of the multilevel scheme is required. The current paper is considered as a follow-up paper to [3,10,11,13]. We refer readers to them for further details. Our proposed method not only settles the issue of how to choose the support, but also improves the accuracy and provides a stopping algorithm similar to an iterative method. Two numerical examples, a Poisson’s problem in 2D and an inhomogeneous Helmholtz-type equation in 3D, are given to validate our proposed method.

2. COMPACTLY SUPPORTED RBFS AND THE DRM

Let \( \mathcal{L} \) be a linear second-order elliptic partial differential operator and let a fundamental solution of \( \mathcal{L} \) be given. We consider the following model boundary value problem (BVP):

\[
\mathcal{L}u(x) = f(x), \quad x \in \Omega, \tag{1}
\]

\[
u(x) = g(x), \quad x \in \partial\Omega, \tag{2}
\]

where \( \Omega \subset \mathbb{R}^d \), \( d = 2,3 \), is a bounded domain with a sufficiently regular boundary \( \partial\Omega \). We assume that BVP \((1),(2)\) has a unique solution \( u \) for any given continuous inhomogeneous term \( f \) and the boundary data \( g \).
Throughout this paper, we exploit the fact that there are excellent efficient robust computational schemes to solve the associated homogeneous problem, using boundary integral representation of the homogeneous solution, say $u_h$. Our focus in this paper is to propose efficient computational schemes to find approximate particular solutions of (1) with $L$ typically being the Laplace or a Helmholtz type operator.

Accordingly, it is a common practice to split the solution $u$ of (1),(2) into a particular solution and homogeneous solution. Let $u = u_p + u_h$, where $u_p$ is a particular solution satisfying the inhomogeneous equation

$$Lu_p(x) = f(x),$$

but does not necessarily satisfy the boundary condition in (2). Then homogeneous solution $u_h$ satisfies

$$Lu_h(x) = 0, \quad x \in \Omega,$$

$$u_h(x) = g(x) - u_p(x), \quad x \in \partial\Omega.$$  

Since $u_h$ satisfies (4),(5) and a fundamental solution of $L$ is known, we can find a suitable equivalent boundary integral equation (BIE) formulation of (4),(5). The resulting BIE formulations contain no domain integration, a marked advantage of solving only an equivalent $(d - 1)$ dimensional equation, see for example [15] and extensive references therein. For our computation, to achieve a mesh free boundary method, we employ the method of fundamental solutions (MFS) [16-18] to solve (4),(5). One may also use robust boundary element or spectral methods to solve boundary integral equations equivalent to (4),(5).

The key issue henceforth is how to compute an approximation to $u_p$ for a general forcing term $f$ in (3). There are various ways of doing so. The DRM has emerged as a promising technique in approximating particular solution $u_p$. The success of the DRM relies on choosing some proper basis functions. We review CS-RBFs approximations from DRM context in rest of this section.

Let $\varphi : \mathbb{R}^+ \to \mathbb{R}$ be a continuous function with $\varphi(0) \geq 0$. For a given $x_i \in \Omega$, we define a function $\varphi_i$ on $\Omega$ by

$$\varphi_i(x) = \varphi(||x - x_i||), \quad x \in \Omega,$$

where $|| \cdot ||$ is the Euclidean norm. Then $\varphi_i$ is called a radial basis function. Instead of globally defined functions, a CS-RBF is a radial basis function with local support. For a discussion of the CS-RBFs, we refer readers to [7-9]. In Table 1, we give a list of CS-RBFs which were constructed by Wendland [8]. It contains the lowest possible degree among all piecewise polynomial CS-RBFs which are positive definite on $\mathbb{R}^d$ for a given order of smoothness. In the current context, we are only interested in the case $d = 2,3$.

<table>
<thead>
<tr>
<th>$d = 2,3$</th>
<th>$\varphi = (1 - r)^2$</th>
<th>$C^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi = (1 - r)^4_+ (4r + 1)$</td>
<td>$C^2$</td>
<td></td>
</tr>
<tr>
<td>$\varphi = (1 - r)^6_+ (35r^2 + 18r + 3)$</td>
<td>$C^4$</td>
<td></td>
</tr>
<tr>
<td>$\varphi = (1 - r)^8_+ (32r^3 + 25r^2 + 8r + 1)$</td>
<td>$C^6$</td>
<td></td>
</tr>
</tbody>
</table>

In Table 1, we used the notation

$$(1 - r)_+^n = \begin{cases} (1 - r)^n, & \text{if } 0 \leq r \leq 1, \\ 0, & \text{if } r > 1. \end{cases}$$

We also note that the radius of support in Table 1 has been normalized to $[0,1]$. 

For practical implementations, we need to rescale the support of \( \varphi \). This can be achieved by using instead the scaled function

\[
\varphi^{[\alpha]}(r) = \varphi\left( \frac{r}{\alpha} \right),
\]

for various values of \( \alpha > 0 \). As in (6), for a given \( x_i \in \Omega \), we define the scaled CS-RBF \( \varphi_i \) on \( \Omega \) as

\[
\varphi^{[\alpha]}_i(x) = \varphi^{[\alpha]}(\|x - x_i\|), \quad x \in \Omega.
\]

For the details of the scaling effect, in data fitting, we refer readers to [4,7,14].

The DRM is based on the assumption that we can find an approximation \( \tilde{f} \) to the inhomogeneous term \( f \) in (3) and that we can obtain an analytical solution \( \tilde{u}_p \) to

\[
\mathcal{L}\tilde{u}_p(x) - \tilde{f}(x).
\]

Then \( \tilde{u}_p \) can be treated as an approximation to a particular solution \( u_p \) of (3). The initial step of the DRM is to approximate \( f(x) \) by using various kinds of basis functions. To avoid the ill-conditioning problem as we indicated in the Introduction, we choose CS-RBFs as basis functions. More precisely, we choose a support parameter \( \alpha \) and a set of uniformly distributed points in \( \Omega \), say \( \{x_j\}_{j=1}^N \), and seek an approximation \( \tilde{f} \) to \( f \) in the form

\[
\tilde{f}(x) := \sum_{j=1}^N a_j \varphi^{[\alpha]}_j(x) = \sum_{j=1}^N a_j \varphi^{[\alpha]}(\|x - x_j\|), \quad x \in \Omega.
\]

(Later in our notation we will indicate the dependence of \( \tilde{f} \) on the support parameter and number of chosen points in \( \Omega \).) The unknown coefficients \( a_j, j = 1, \ldots, N \), are determined by forcing the interpolatory conditions

\[
\tilde{f}(x_i) = f(x_i), \quad 1 \leq i \leq N.
\]

The resulting linear system

\[
\sum_{j=1}^N a_j \varphi^{[\alpha]}_j(x_i) = f(x_i), \quad 1 \leq i \leq N,
\]

is well posed if the interpolation matrix

\[
A_{\varphi^{[\alpha]}} = \begin{bmatrix} \varphi^{[\alpha]}(\|x_i - x_j\|) \end{bmatrix}_{1 \leq i,j \leq N}
\]

is nonsingular. Since CS-RBFs are positive definite, the induced matrix \( A_{\varphi^{[\alpha]}} \) in (13) is also positive definite. This ensures the solvability of (12). It is useful to note that \( A_{\varphi^{[\alpha]}} \) will be dense or sparse depending on the support parameter \( \alpha \).

Once \( \tilde{f} \) in (9) has been established, using (10), an approximation \( \tilde{u}_p \) (depending on \( N \) and \( \alpha \)) to a particular solution \( u_p \) of (1) can be written as

\[
\tilde{u}_p(x) = \sum_{j=1}^N a_j \Phi_j^{[\alpha]}(x), \quad x \in \Omega,
\]

where \( \Phi_j^{[\alpha]} \) is the solution of

\[
\mathcal{L}\Phi_j^{[\alpha]}(x) = \varphi_j^{[\alpha]}(x), \quad x \in \Omega, \quad j = 1, \ldots, N.
\]

One of the key steps in the DRM is the possibility of solving (15) analytically. Since the forcing terms in (15) are radially dependent functions on \( \Omega \), one may expect analytical solvability of (15)
for operators $L$ that are radially and translationally invariant. For such operators if we denote $L_r$ to be the radial part of $L$, then finding an analytical representation of $\tilde{u}_p$ in (14) is equivalent to finding the analytical solution $\Phi^{[\alpha]}(r)$ of the radial differential equation

$$L_r \Phi^{[\alpha]}(r) = \varphi^{[\alpha]}(r), \quad r \geq 0,$$

(16)

where $\varphi^{[\alpha]}$ is given by (7) with typical representations of $\varphi$ as in Table 1. Explicit analytical representation of $\Phi^{[\alpha]}(r)$ for the special case $L = \Delta$, the Laplacian, was derived in [10,11]. Recently in [13], we derived $\Phi^{[\alpha]}(r)$ for the Helmholtz-type operators $L = \Delta - s^2I$ (for any real or complex number $s^2$) in three dimensions.

In particular, for the CS-RBF $\varphi = (1 - r)^4 (4r + 1)$, we have the following explicit formulas for $\Phi^{[\alpha]}(r)$:

- $d = 2$, $L = \Delta$ (see [10])

$$\Phi^{[\alpha]}(r) = \begin{cases}
\frac{r^2}{4} - \frac{5r^4}{8\alpha^2} + \frac{4r^5}{5\alpha^3} - \frac{5r^6}{12\alpha^4} + \frac{r^7}{49\alpha^5}, & r \leq \alpha, \\
\frac{529\alpha^2}{5880} + \frac{\alpha^2}{14} \ln \left( \frac{r}{\alpha} \right), & r > \alpha;
\end{cases}$$

(17)

- $d = 3$, $L = \Delta$ (see [11])

$$\Phi^{[\alpha]}(r) = \begin{cases}
\frac{r^2}{6} - \frac{r^4}{2\alpha^2} + \frac{2r^5}{3\alpha^3} - \frac{5r^6}{14\alpha^4} + \frac{r^7}{14\alpha^5}, & r \leq \alpha, \\
\frac{\alpha^2}{14} - \frac{\alpha^3}{42r^5}, & r > \alpha;
\end{cases}$$

(18)

- $d = 3$, $L = \Delta - s^2I$ (see [13])

$$\Phi^{[\alpha]}(r) = \begin{cases}
s(2B + q(0)) + q'(0), & r = 0, \\
\frac{[Ae^{-sr} + Be^{sr} + q(r)]}{r}, & 0 < r \leq \alpha, \\
Ce^{-sr}, & r > \alpha,
\end{cases}$$

(19)

where

$$A = -[B + q(0)], \quad B = -\frac{e^{-s\alpha} [q'(\alpha) + sq(\alpha)]}{2s}$$

(20)

$$C = B \left( e^{2s\alpha} - 1 \right) + q(\alpha)e^{s\alpha} - q(0),$$

(21)

and

$$q(r) = -\frac{480}{s^6\alpha^3} - \frac{2880}{s^8\alpha^5} + \left( \frac{60}{s^4\alpha^2} + \frac{1800}{s^6\alpha^4} - \frac{1}{s^2} \right) r - \left( \frac{240}{s^6\alpha^3} + \frac{1440}{s^8\alpha^5} \right) r^2$$

$$+ \left( \frac{10}{s^2\alpha^2} + \frac{300}{s^4\alpha^4} \right) r^3 - \left( \frac{20}{s^4\alpha^3} + \frac{120}{s^6\alpha^5} \right) r^4 + \frac{15}{s^2\alpha^4} r^5 - \frac{4}{s^2\alpha^5} r^6.$$}

(22)

In fact, the formula $\Phi^{[\alpha]}(r)$ for the Helmholtz-type operators in (19)–(21) holds for all CS-RBFs and only the representation of $q$ in (22) differs for various choices of CS-RBFs. These particular solutions for Helmholtz-type operators using CS-RBFs in 3D are expected to have extensive applications in solving three dimensional time-dependent problems using boundary integral and fundamental solution methods. For details, we refer to [13]. Further, we expect, depending
on the applications, for nonstandard radially invariant operators $L$ one may be able to derive representation of the solution $\Phi^{(a)}$ of (16).

Thus, the main computational cost of finding approximate particular solutions of (1) is in computing the unknown coefficient vector $a = [a_1, \ldots, a_N]^T$ in (10) by solving the linear system in (12), given by $A_{\alpha,\alpha}a = f$, where $f = [f(x_1), \ldots, f(x_N)]^T$.

Clearly, the quality of the approximation of $f$ by $\hat{f}$ (depending on $N$ and $\alpha$) and the stability of the corresponding computing process determine the accuracy of the approximate particular solution. Following [14] and references therein, for sufficiently smooth functions $f$, the measure of quality of approximation $\hat{f}$ depends on $h_\alpha$, where

$$h_\alpha = \frac{1}{\alpha} \sup_{x \in \Omega} \min_{1 \leq j \leq N} \|x - x_j\|,$$  \hfill (23)

while the sensitivity of the stability of the interpolation process depends on $q_\alpha$, where

$$q_\alpha = \frac{1}{\alpha} \min_{1 \leq k \leq j \leq N} \|x_k - x_j\|.$$ \hfill (24)

In particular, ideally we would like $h_\alpha$ to be as small as possible and $q_\alpha$ as large as possible. Further, from the computational point of view, we require the interpolation matrix $A_{\alpha,\alpha}$ to be of moderate size and not dense, i.e., the number of points $N$ not very large and $\alpha$ as small as possible. It is clear that it will not be possible to satisfy all the above requirements for a single choice of the cut-off parameter $\alpha$. Also, for large scale problems, we need to take $N$ sufficiently large.

The best way to tackle the above issue, to a certain extent, is to consider a multi-cut-off level scheme by choosing various size interpolation point sets and corresponding cut-off parameters. We propose two multilevel algorithms in the next section following ideas in [14].

3. MULTILEVEL SCHEMES

Let $D_N = \{x_i\}_{i=1}^N$ be a set of interpolation points. Here the number of points $N$ is large enough to cover $\Omega$ with $h_1$ given by (23) is sufficiently small. Next, we subdivide $D_N$ into a sequence of evenly distributed point sets

$$D_1 \subset D_2 \subset \cdots \subset D_k \subset \cdots \subset D_L = D_N,$$

where $D_k = \{x_i\}_{i=1}^{N_k}$ with appropriately chosen $N_k$, $k = 1, \ldots, L - 1$ and $N_L = N$. For each $k = 1, \ldots, L$, the parameter $N_k$ depends on the choice of a cut-off parameter $\alpha_k$, where

$$\alpha_1 > \alpha_2 > \cdots > \alpha_k > \cdots > \alpha_L$$

is a chosen set of cut-off parameters. The choice of $\alpha_k$ and $N_k$, $k = 1, \ldots, L$, depend on the required minimal accuracy of the approximation and the size and sparsity constraints of the interpolation matrix $A_{\alpha,\alpha}$). Due to the complex nature of requirements mentioned in the last section, there is no perfect choice of such parameters satisfying all the constraints. The multilevel scheme we propose from the DRM point of view is to first obtain a crude approximate solution of (1), (2) in the first few levels with few interpolation points but large scaling factors. For example, we may try various choices until the first level produces an approximate solution with one decimal accuracy. We then refine the solution by approximating the residues.

In [14], a thinning algorithm has been devised to produce a sequence of evenly distributed subsets $D_k$, $k = 1, \ldots, L - 1$, of interpolation points. However, the filtering process of the thinning algorithm is quite tedious. For purely surface fitting, the data points may be collected from the field and the thinning algorithm is necessary for implementing the multilevel method.
For solving a partial differential equation, the interpolation points are normally selected in the domain. To this end, we use a quasi-Monte Carlo method [19] to generate a sequence of quasi-random points which also ensures that the interpolation points are uniformly distributed at each level. For computation, we use the subroutine SOBSEQ [20] to generate quasi-random interpolation points and translate to cover the domain \( \Omega \).

The fundamental idea of the multilevel scheme is to capture the main features of \( f \) in \( \hat{f} \) in the first few levels with few interpolation points but large scaling factors. Small details are added on in the later steps which consist of large number of interpolation points but small scaling factors.

### 3.1. Multilevel Algorithm 1

Our first multilevel scheme approach to find an approximate solution of (3) can be described as follows.

For \( k = 1, \ldots, L \), with \( \alpha_k \) being the scaling factor for \( D_k \), we set

\[
\hat{f}^k(x) = \sum_{j=1}^{N_k} c_j^{(k)} \phi_j^{(\alpha_k)} \left( \frac{\|x - x_j^{(k)}\|}{\alpha_k} \right), \quad x_j^{(k)} \in D_k
\]

and at level \( k \) we choose the approximate particular solution of (1) as

\[
\hat{u}_p^k(x) = \sum_{j=1}^{N_k} c_j^{(k)} \Phi_j^{(\alpha_k)}(x), \quad x \in \Omega,
\]

where \( \Phi_j^{(\alpha_k)} \) is a solution of

\[
\mathcal{L} \Phi_j^{(\alpha_k)}(x) = \phi_j^{(\alpha_k)}(x), \quad x \in \Omega, \quad j = 1, \ldots, N_k.
\]

The solutions \( \Phi_j^{(\alpha_k)} \), \( j = 1, \ldots, N_k \), \( k = 1, \ldots, L \), can be computed using analytical formulas such as described in (17)-(22). For \( k = 1 \), the coefficients \( c_j^{(1)} \), \( j = 1, \ldots, N_1 \), in (25) and (26) are determined by

\[
\hat{f}^1(x_i) = f(x_i), \quad 1 \leq x_i \leq N_1.
\]

and for \( k = 2, \ldots, L \) the coefficients \( c_j^{(k)} \), \( j = 1, \ldots, N_k \), in (25) and (26) are computed using the interpolatory constraints

\[
\hat{f}^k(x_i) = f(x_i) - \sum_{j=1}^{k-1} \hat{f}^j(x_i), \quad 1 \leq x_i \leq N_k.
\]

Consequently, at each level \( k = 1, \ldots, L \), the inhomogeneous function \( f \) is approximated by \( \sum_{i=1}^k \hat{f}^k \). Clearly \( f(x) = \lim_{L \to \infty} \sum_{k=1}^L \hat{f}^k(x) \), and \( f^k(x) \to 0 \) as \( k \to \infty \) for \( x \in \Omega \).

At the first level, we choose the support value \( \alpha_1 \) high and the number of points \( N_1 \) in \( D_1 \) low and obtain the unknown coefficient vector \( c^{(1)} = [c_1^{(1)}, \ldots, c_{N_1}^{(1)}]^T \) by solving the \( N_1 \times N_1 \)-dense system

\[
A_{\phi^{(\alpha_1)}} c^{(1)} = f^{(1)},
\]

where \( f^{(1)} = [f(x_1), \ldots, f(x_{N_1})]^T \). For subsequent levels \( k = 2, \ldots, L \), we are actually interpolating the residual of the previous levels. That is, for \( k = 2, \ldots, L \), to compute the vector \( c^{(k)} = [c_1^{(k)}, \ldots, c_{N_k}^{(k)}]^T \), we solve the \( N_k \times N_k \) finite-dimensional systems of the form

\[
A_{\phi^{(\alpha_k)}} c^{(k)} = f^{(k)}
\]
with the first $N_{k-1}$ entries of $f^{(k)}$ being zeros and the remaining $N_k - N_{k-1}$ are given by (29). As the level increases we decrease the support value and increase the number of interpolation points. Consequently, we solve a sequence finite dimensional systems with increasing dimension as well as sparsity. For example, see Tables 2 and 3 in Section 4. Due to excellent properties of the CS-RBFs, the resulting systems are symmetric and positive definite (SPD). Hence, the sparse SPD systems can be solved efficiently using for example conjugate gradient type methods.

We then continue our algorithm by setting the approximate particular solution $\tilde{u}_p^N$ of $(1)$ as

$$\tilde{u}_p^N = \sum_{k=1}^L \tilde{u}_p^k,$$

with $u_p^k$, $k = 1, \ldots, L$, given by (26) and $N = N_L$, the number of chosen interpolation points in the final level. To show that $\tilde{u}_p^N$ is indeed a good approximation particular solution of $(1)$, we observe first from (32), (26), (27), and (25) that

$$\mathcal{L} \tilde{u}_p^N = \mathcal{L} \left( \sum_{k=1}^L \tilde{u}_p^k \right) = \sum_{k=1}^L (\mathcal{L} \tilde{u}_p^k) = \sum_{k=1}^L \tilde{f}^k.$$

Since $\sum_{k=1}^L \tilde{f}^k \to f$ as $L \to \infty$, we have $\tilde{u}_p^N \to u_p$ as $N \to \infty$, where $u_p$ is a particular solution of $(1)$. In general, the particular solution in $(1)$ is not unique, and hence, our above algorithm yields approximate particular solutions $u_p^N$ converging to a 'particular' particular solution of $(1)$. Using the approximate particular solution (32), the final step in our algorithm is to compute an approximate solution $u_N^N$ of the associate approximate homogeneous problem of $(1), (2)$ given by

$$\mathcal{L} u_N^N(x) = 0, \quad x \in \Omega,$$

$$u_N^N(x) = g(x) - \tilde{u}_p^N(x), \quad x \in \partial \Omega.$$

One may use boundary integral methods or MFS to solve (33), (34). Finally, we take $u^N = u_h^N + \tilde{u}_p^N$ as an approximation to the unique solution $u$ of $(1), (2)$. One unresolved issue in the above algorithm is how many levels $L$ to be chosen.

For practical purposes, one may actually be interested in finding approximate solutions of $(1), (2)$ to certain specified accuracy, say $TOL$. Assuming that one may find approximate homogeneous solutions to any specified accuracy using, for example, robust boundary integral software packages, we need a multilevel algorithm for finding approximate particular solutions in such a way that the combined approximate solutions of the inhomogeneous problem $u_N^N$ satisfy the $TOL$ accuracy. The above multilevel algorithm requires a priori the choice of number of levels $L$ and computation of particular solutions at all the levels $k = 1, \ldots, L$ before computing approximate solutions of the homogeneous problem $(33), (34)$. Consequently, in our Multilevel Algorithm 1, we need a stopping criteria for choosing $N$ depending on $TOL$.

One approach to tackle this problem is to observe that, since $\tilde{f}^k(x) \to 0$ as $k \to \infty$ for $x \in \Omega$, we expect that, $\|c^{(k)}\| \to 0$ as $k \to \infty$. Hence, from (26), we expect $\|\tilde{u}_p^k\|_\infty \to 0$ as $k \to \infty$. So one may think of using the criteria that we proceed with the above Multilevel Algorithm 1 up to level $k$ and choose $N = N_k$ as the stopping criteria where $N_k$ is such that $\|\tilde{u}_p^k\|_\infty < TOL$. However, this criteria may not be robust in general: for a finite level $k$, $\tilde{u}_p^k$ depends both on the decaying coefficient vector $c^{(k)}$ as well as the solutions $\Phi_j^{[\alpha_1]}$ of (27). But for all $j = 1, \ldots, N_k$, $\Phi_j^{[\alpha_1]} + C$ is also a solution of (27) for any constant $C$. So, for example, one may have chosen $\Phi_j^{[\alpha_1]} + 1/\|c^{(k)}\|$ as a solution of (27) there by making the criteria for a $\|\tilde{u}_p^k\|_\infty < TOL$ for a finite $k$ not robust.

In the next section, we propose a robust stopping criteria based multilevel algorithm. The trade off is that some additional computation cost is added.
3.2. Multilevel Algorithm 2

We propose a multilevel approach by decomposing the exact unique solution of (1),(2) into a series of solutions of simpler inhomogeneous equations based on the Multilevel Algorithm 1. Our aim is to devise a multilevel CS-RBFs based computational scheme to find approximate solutions of (1),(2) satisfying a given error tolerance TOL.

Let \( v_1 \) be the unique solution of

\[
L v_1(x) = f(x), \quad x \in \Omega, \tag{35}
\]
\[
v_1(x) = g(x), \quad x \in \partial \Omega, \tag{36}
\]

and, for each \( k = 2, 3, \ldots \), let \( v_k \) be the unique solution of

\[
L v_k(x) = f^k(x), \quad x \in \Omega, \tag{37}
\]
\[
v_k(x) = 0, \quad x \in \partial \Omega. \tag{38}
\]

In (35) and (37), \( f^k \) is as defined in (25) with interpolatory constraints given by (28) for \( k = 1 \) and by (29) for \( k = 2, 3, \ldots \). Since \( f(x) = \sum_{k=1}^{\infty} f^k(x), \quad x \in \Omega, \) it is easy to see that the unique solution \( u \) of (1),(2) can be written as

\[
u(x) = \sum_{k=1}^{\infty} v_k(x), \quad x \in \Omega. \tag{39}
\]

Further, since \( \| f^k \|_{\infty} \to 0 \) as \( k \to \infty \), we have \( \| v^k \|_{\infty} \to 0 \) as \( k \to \infty \) and \( \{ \| v^k \|_{\infty} \} \) is a strictly monotonically decreasing sequence of real numbers.

We set an approximate solution of (1),(2) to be

\[
u_N(x) = \sum_{k=1}^{L} v_k(x), \quad x \in \Omega, \tag{40}
\]

where \( L \) is to be chosen such that \( N = N_L \) is the number of interpolation points in \( D_L \) and that \( \| u - u_N \|_{\infty} = \| \sum_{k=L+1}^{\infty} v_k \|_{\infty} < TOL \). Using the properties of the unique solution \( v_k \) of (37),(38), the stopping criteria of finding \( L \) can be achieved (approximately) by looking for the minimum iteration level \( L \) such that \( \| v^L \|_{\infty} < TOL \). (Perhaps one may also choose \( \| v^L \|_{\infty} < (TOL)^2 \) or \( \| v^L \|_{\infty} < c \cdot TOL \) for some constant \( c \ll 1 \) as a stronger stopping criteria.) To compute approximate solutions \( v^k, \quad k = 1, \ldots, L \), we proceed as follows.

For \( k = 1 \), we write \( v_1 = \hat{u}_p^1 + u_h^1 \), where \( \hat{u}_p^1 \) is a particular solution of (35) computed using representation (26) and (28), and \( u_h^1 \) is the approximate solution of the homogeneous problem

\[
Lu_h^1(x) = 0, \quad x \in \Omega, \tag{41}
\]
\[
u_h^1(x) = g(x) - \hat{u}_p^1(x), \quad x \in \partial \Omega. \tag{42}
\]

The homogeneous BVP (41),(42) can be solved, for example, using robust boundary integral or MFS approach with high accuracy. This will lead to solving a finite-dimensional system with a dense matrix \( M_L \) (independent of the boundary data). The procedure of solving this system should involve finding first an LU factorization of \( M_L \) (main computational cost). Then we compute an approximate solution \( u_h^1 \) using the LU factorization with cheaper appropriate matrix vector multiplications.

For \( k = 2, \ldots, L \), we write \( v_k = \hat{u}_p^k + u_h^k \), where \( \hat{u}_p^k \) is a particular solution of (37) computed through (26) and (29), and \( u_h^k \) is the approximate solution of the homogeneous problem

\[
Lu_h^k(x) = 0, \quad x \in \Omega, \tag{43}
\]
\[
u_h^k(x) = -\hat{u}_p^k(x), \quad x \in \partial \Omega. \tag{44}
\]
Using the LU factorization of $M_L$, the approximate solution $\hat{u}_k^p$ at each level $k = 2, \ldots, L$ can be computed easily with just appropriate matrix vector multiplications involving the $\hat{u}_p$.

It is useful to note that compared to Algorithm 1, described in Section 3.1, the above algorithm involves solving in addition the homogeneous problems at levels $k = 2, \ldots, L$. Since, in practice, the maximum number of levels $L$ is not expected exceed 10, the additional matrix vector multiplications computational cost involved in Algorithm 2 is justified if stopping criteria is an important issue for certain practical problems involving adaptive type coding. For many simple test problems, perhaps Algorithm 1 may be sufficient.

### 4. NUMERICAL RESULTS

In this section, we demonstrate the effectiveness of our proposed multilevel schemes by computing approximate solutions of a 2D Poisson problem and a 3D modified Helmholtz equation with $\text{TOL} = 10^{-3}$.

**EXAMPLE 1.** Consider the Poisson’s problem

\[
\begin{align*}
\Delta u(x, y) &= f(x, y), \quad \text{in } \Omega, \\
u(x, y) &= g(x, y), \quad \text{on } \partial \Omega,
\end{align*}
\]

where $\Omega \cup \partial \Omega = \{(x, y) \in \mathbb{R}^2 : (x - 1.5)^2 + (y - 1.5)^2 \leq 0.25\}$. For testing purposes, we chose $f$ and $g$ in such a way that the exact solution of (45),(46) is

\[
u(x, y) = \sin \frac{\pi x}{6} \sin \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4}, \quad (x, y) \in \Omega.
\]

A plot of the exact solution is in Figure 1. (For convenience of viewing, we extended the graphs of functions involved in this test example to the square $[1, 2] \times [1, 2]$.)
Choice (47) is possible if the boundary data $g$ in (46) is the same as in (47) and if the inhomogeneous term $f(x, y)$ is given by

$$f(x, y) = -\frac{75\pi^2}{144} \sin \frac{\pi x}{6} \sin \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4} + \frac{7\pi^2}{12} \cos \frac{\pi x}{6} \cos \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4} + \frac{15\pi^2}{8} \sin \frac{\pi x}{6} \cos \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \cos \frac{5\pi y}{4}.$$

As shown in Figure 2, the forcing term $f(x, y)$ has a relatively large fluctuation in the domain, a reason for choosing the nonstandard test solution.

![Figure 2. The inhomogeneous term $f(x, y)$ in (46).](image)

For computation, we chose the CS-RBF $\varphi(r) = (1 - r)^{4}(4r + 1)$. Using the quasi-Monte Carlo based subroutine SOBSEQ [20], we generated $N = 500$ quasi-random points in the disk $D = \{(x, y) : (x - 1.5)^2 + (y - 1.5)^2 \leq 9/16\}$ which contains $\Omega \cup \partial \Omega$. Following the notations in Section 3, we chose four levels

$$\alpha_1 = 0.8, \quad \alpha_2 = 0.5, \quad \alpha_3 = 0.3, \quad \alpha_4 = 0.18,$$

and

$$N_1 = 30, \quad N_2 = 150, \quad N_3 = 300, \quad N_4 = 500$$

and for $k = 1, 2, 3, 4$, $D_k$ consisting of first $N_k$ points from the generated quasi-random points in $D$ (see Figure 3). (Our choice of extending the interpolation points to outside the physical domain improves the accuracy in evaluating the particular solutions. In this way, the boundary points will have enough supports as the interior points.)

For $k = 1, 2, 3, 4$ the sparsity structure (with nonzero entries $n_{zk}$) of the resulting $N_k \times N_k$ matrix $A_{\varphi^{(k)}}$ (see (30) and (31)) is given in Table 2.

All the graphs below for the test Example 1 were produced on a $25 \times 25$ uniform grid in $[1, 2] \times [1, 2]$. The numerical results outside the domain $\Omega \cup \partial \Omega$ are set to 0. For each $k = 1, 2, 3, 4$, we solved the SPD system $A_{\varphi^{(k)}} c^{(k)} = f^{(k)}$ (see (30) and (31)) using the IMSL library SPD solver DLSLDX (PC version).
Figure 3. Quasi-random interpolation points at each level.

Table 2. Sparsity pattern of the interpolation matrix $A_{\phi|n_k}$, $k = 1, 2, 3, 4$.

<table>
<thead>
<tr>
<th>$N_k$</th>
<th>$n_{zk}$</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>281</td>
<td>31.22</td>
</tr>
<tr>
<td>150</td>
<td>3606</td>
<td>16.02</td>
</tr>
<tr>
<td>300</td>
<td>5979</td>
<td>6.64</td>
</tr>
<tr>
<td>500</td>
<td>6693</td>
<td>2.67</td>
</tr>
</tbody>
</table>

The profile of the interpolation error of the forcing term $e_k = f - \sum_{i=1}^{k} f^k$ at each level $k = 1, 2, 3, 4$ are given in Figure 4. Our actual interest (from both multilevel schemes in Section 3) is in the quality of the particular solution $\hat{u}^k_p$, $k = 1, 2, 3, 4$, computed using (26) and $c^{(k)}$. These are given in Figure 5.

As expected, from Figures 4 and 5, we observe that $\|e_k\|_{\infty}$ and $\|\hat{u}^k_p\|_{\infty}$ get smaller as the level increases. We demonstrate the need for both our multilevel algorithms described in Section 3, depending on the requirement of robust stopping criteria to achieve the desired accuracy $TOL = 10^{-3}$.

Suppose that we wanted to compute an approximate solution of (45), (46) up to $TOL$ accuracy using our Multilevel Algorithm 1. Since $\|\hat{u}^4_p\|_{\infty} > TOL$ it is not clear whether we need to do computation with more levels or following arguments at the end of Section 3.1, these four levels may be sufficient to reach the desired $TOL$ accuracy.

As a check on Multilevel Algorithm 1, we stopped with the computed four levels of particular solutions and summed up these as in (32) to get an approximate particular solution $\hat{u}^N_p$ of (45). Then we computed an approximate solution $u^N_h$ of the associated homogeneous problem (33), (34) (with $L = \Delta$) using the MFS [16–18] with 40 evenly distributed collocation points on $\partial \Omega$ and
Figure 4. The profile of error function $e_k$ at level $k = 1, 2, 3, 4$.

Figure 5. The profile of the particular solution $u_p^k$ at level $k = 1, 2, 3, 4$. 
39 source points on a circle of radius 10. The accuracy of our computed approximate solution \(u^N = \hat{u}_p^N + \hat{u}_p^N\) of (45),(46) plotted in Figure 6 shows that we reached the required accuracy \(\|u - u^N\|_{\infty} < \text{TOL}\). Thus, demonstrating our Multilevel Algorithm 1 is efficient in obtaining approximate solutions, but does not provide robust stopping criteria of the number of levels to be chosen.

Next we used our Multilevel Algorithm 2 using the computed approximate particular solutions \(\hat{u}_p^k, k = 1, 2, 3, 4\) and the MFS strategy. Following notations in Section 3.2, we computed \(v^k\) and found that \(\|v^k\|_{\infty} > \text{TOL}\) for \(k = 1, 2, 3\) and then computed \(v^4\). Since \(\|v^4\|_{\infty} < \text{TOL}\), we stopped further multilevel computation. The plots of \(v^k, k = 1, 2, 3, 4\) are given in Figure 7.

We summed up the solutions \(v^k, k = 1, 2, 3, 4\) as in (40) to obtain an approximate solution \(u_N\). The resulting error function \(e_N(x,y) = |u(x,y) - u_N(x,y)|\) in Figure 8, demonstrates the effectiveness of our Multilevel Algorithm 2.

**EXAMPLE 2.** We consider the following 3D Helmholtz-type problem (with \(s^2 = 1\)):

\[
(\Delta - s^2 I) u = f, \quad \text{in } \Omega, \\
u = g, \quad \text{on } \partial \Omega,
\]

where the domain (see Figure 0) \(\Omega \subset \mathbb{R}^3\) is given by \(\Omega = \{(x, y, z) \in \mathbb{R}^3 : H(x, y, z) < 1\}\) with \(H(x, y, z) = \min\{(4x - 3/4)^2, (4z + 3/4)^2\} + (4y)^2 + (4z)^2\).

For numerical test purposes, we chose the inhomogeneous term \(f\) in (48) and the boundary function \(g\) in (49) so that \(u(x, y, z) = \cosh(x + y + z)\) is the exact solution of (48),(49). More, precisely on \(\partial \Omega\), \(g(x, y, z) = \cosh(x + y + z)\) (see Figure 10) and on \(\Omega\), \(f(x, y, z) = 2 \cosh(x + y + z)\). A cross-section plot of \(f\) along \(z = 0\), i.e., \(f(x, y, 0)\) with \(H(x, y, 0) < 1\) is in Figure 11.

We used the CS-RBF \(\varphi^{[p]}(r) = (1 - r/\alpha)^p(4(r/\alpha) + 1)\) and first generated \(N = 500\) quasi-random points in \(\Omega\). We chose four levels

\[
\alpha_1 = 1.0, \quad \alpha_2 = 0.7, \quad \alpha_3 = 0.5, \quad \alpha_4 = 0.3
\]

and

\[
N_1 = 10, \quad N_2 = 30, \quad N_3 = 120, \quad N_4 = 500
\]
Figure 7. The profile of Multilevel Algorithm 2 solutions $v_k$ at levels $k = 1, 2, 3, 4$. 
Figure 8. The profile of Multilevel Algorithm 2 error function $e_N$.

Figure 9. Domain $\Omega \subset \mathbb{R}^3$.

Figure 10. Exact solution $u = g$ on $\partial \Omega$. 
and, for $k = 1, 2, 3, 4$, $D_k$ consisted of first $N_k$ points from the generated quasi-random points as shown in Figure 12.

We used MATLAB on a multiuser workstation for computation. For $k = 1, 2, 3, 4$, the sparsity structure (with nonzero entries $n_{zk}$) of the resulting $N_k \times N_k$ matrix $A_{\varphi^{(a_k)}}$ (see (30) and (31)) is given in Table 3.

![Figure 11. Inhomogeneous term $f = 2u$ on $\Omega$ along $z = 0$.](image1)

![Figure 12. Quasi-random points $D_k \subset \mathbb{R}^3$, $k = 1, 2, 3, 4$ for interpolation.](image2)

<table>
<thead>
<tr>
<th>$N_k$</th>
<th>$n_{zk}$</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>30</td>
<td>882</td>
<td>98</td>
</tr>
<tr>
<td>120</td>
<td>11560</td>
<td>80.28</td>
</tr>
<tr>
<td>500</td>
<td>100672</td>
<td>40.27</td>
</tr>
</tbody>
</table>
Figure 13. Particular solution $u_k^*$ on $\Omega$ along $z = 0$ at level $k = 1, 2, 3, 4$.

Figure 14. Approximate solution $v_k^*$ on $\Omega$ along $z = 0$ at level $k = 1, 2, 3, 4$. 
We used biconjugate gradient method (with accuracy $10^{-10}$) to solve the SPD matrix problem $A(x) = f(x)$ (see (30) and (31)) at each level $k = 1, 2, 3, 4$. Next using representations (26) and (19) for $k = 1, 2, 3, 4$, we computed approximate particular solutions $\hat{u}^k_p$ of the 3D Helmholtz-type equation (48). The particular solutions on $\Omega$ along $z = 0$ are given in Figure 13. The decaying behaviour of $\hat{u}^k$ was observed throughout the domain $\Omega$.

As mentioned in Section 3.1 and demonstrated in Example 1, the decaying property of the particular solution at the fourth level is not sufficient to determine a stopping criteria. With desired accuracy level TOL of the approximate solution of (48), (49), we continued our computation using Multilevel Algorithm 2. As described in Section 3.2, for $k = 1, 2, 3$, using the approximate particular solutions $\hat{u}^k_p$, and MFS [16-18] for the associated homogeneous problem with 50 evenly distributed collocation points on $\partial\Omega$ and 50 source points on the surface of a larger domain (similar to the shape of $\Omega$), we computed approximate solutions $v^k$ and found that $\|v^k\|_{\infty} > TOL$. Based on our stopping criteria, we similarly computed $v^4$ and stopped further computation since $\|v^4\|_{\infty} < TOL$. The behaviour of the solutions $v^k$, $k = 1, 2, 3, 4$ as shown in Figure 14 along $z = 0$ was observed throughout the domain $\Omega$.

Finally, for each $k = 1, 2, 3, 4$, we computed $u_N^{(k)} = \sum_{i=1}^{k} v^i$ and took $u_N = u_N^{(4)}$ as the approximate solution and found that $\|u-u_N\|_{\infty} < TOL$. The plots of the absolute error function $e_k = |u - \sum_{i=1}^{k} u_N^{(k)}|$, $k = 1, 2, 3, 4$ on $\Omega$ along $z = 0$ in Figure 15 demonstrates the accuracy the Multilevel Algorithm 2. Computation of the numerical solutions of the 3D inhomogeneous Helmholtz-type problem took only a few seconds on a standard multiuser workstation. This demonstrates the efficiency of our scheme.

5. CONCLUSIONS

We proposed two multilevel compactly supported radial basis dual reciprocity algorithms to alleviate the difficulty of solving large dense systems associated with the standard radial functions.
boundary only based computational schemes for some inhomogeneous elliptic problems. We demonstrated the algorithms for a 2D Poisson's problem and a 3D Helmholtz-type equations using recently derived particular solutions. Our future work on the multilevel schemes will be concerned with stability and convergence analysis, and an adaptive approach for choosing multilevel CS-RBF cut-off parameters.

REFERENCES