

Recent developments in the dual reciprocity method using compactly supported radial basis functions

C.S. Chen¹, M.A. Golberg² and R.A. Schaback³

¹*Department of Mathematical Sciences*

University of Nevada, Las Vegas, U.S.A.

²*517 Bianca Bay Street, Las Vegas, NV 89144, U.S.A.*

³*Institute of Numerical and Applied Mathematics*

University of Göttingen, Germany

Abstract

We survey some recent developments of the application of compactly supported radial basis functions (CS-RBFs) in the context of the dual reciprocity method. Using the CS-RBFs as the main tool for approximating the right hand side of a given partial differential equation, we further introduce a number of numerical techniques so that a large class of partial differential equations can be solved numerically. Due to the virtue of compact support, the CS-RBFs are very promising for solving large scale and high dimensional problems. In this survey paper, we summarize a collection of closed-form particular solutions for various differential operators. In particular, some of them are new and preliminary numerical results are also provided in this paper. We also point out a few minor errors in previous publications on CS-RBFs and offer the necessary corrections. A number of proposals for future research using CS-RBFs are also suggested.

1 Introduction

In the numerical solution of partial differential equations (PDEs), finite difference and finite element methods (FDM, FEM) are well established techniques for solving various science and engineering problems. Despite their many attractive features for solving PDEs, the FDM usually involves a rectangular grid system, which makes it very difficult to model irregular domains. Although the FEM is more flexible for generating a grid network, it is still a non-trivial task for complicated domains. The domain discretization is often the most time consuming part of the solution process and is far from being fully automated, particularly in 3D. One method to alleviate this difficulty is to use the boundary element method (BEM), which requires only boundary discretization rather

than the domain discretization as mentioned above. In this respect, the efficiency of the BEM is significantly improved over its counterpart FDM and FEM. However, every improvement of a numerical method always comes with other drawbacks of its own. The reduction of the domain discretization to the boundary comes with some price. One of the disadvantages of employing the BEM is that the fundamental solution of the given differential operator is required. This, in principle, means only homogeneous linear differential equations can be solved by the BEM. There are various ways to extend the applicability of the BEM to other types of partial differential equations. As a result, a nonhomogeneous term will somehow present in the right hand side of the reformulated differential equation. The BEM loses its attractiveness when the differential equation is nonhomogeneous due to the fact that domain discretization and domain integration are required.

During the past two decades, much effort has been devoted to dealing with this issue in the BEM community. One of the most widely used methods to transfer domain integrals to the boundary is the dual reciprocity method (DRM) introduced by Nardini and Brebbia [40] in 1982. During the past decade, the application of the DRM in the BEM literature has grown at a rapid pace due in large part to its unique ability to alleviate domain integration when the nonhomogeneous term is involved. The success of the DRM largely depends on how accurately the nonhomogeneous term can be approximated. The early development of the DRM during the period of 1982-1990 was summarized in the book of Partridge et al. [41] where the ad-hoc basis function $1+r$ was exclusively employed for the approximation of the nonhomogeneous term. The theory of radial basis functions (RBFs) was later introduced mainly by Golberg and Chen [19] to replace $1+r$ in which it was recognized that $1+r$ is just a special type of RBF. Since then many of the important papers in the DRM literature have focused on the investigation of the effect of choosing different RBFs [7, 39]. The theoretical development [23] of the DRM using RBFs has put the DRM on a firm mathematical foundation. The follow-up numerical results have confirmed that a good choice of RBFs improves the accuracy and efficiency of the DRM. Consequently, the DRM has been recognized and widely accepted as a reliable numerical method in transferring the domain integral to the boundary in the BEM community.

Another important advance of the DRM was the discovery of closed-form particular solutions for Helmholtz-type operators using RBFs [7, 39]. As we shall see later, so far the task has only been possible by choosing polyharmonic splines or compactly supported RBFs (in 3D) as basis functions. This immediately opens a new research direction for more effectively solving time-dependent problems since most of these types of problems can be reduced to solving a sequence of Helmholtz-type equations via Laplace transform or finite difference schemes [8, 22]. This new development gives strong indication that RBFs provide a rich and flexible class of basis functions that can be implemented so that the derivation of close-form particular solutions for various differential operators is possible. In the past, in the BEM literature the DRM was only applied to the case when the dominant differential operator was kept as the Laplace or bi-harmonic operators. This was primarily due to the difficulty in obtaining close-form particular solutions for other differential operators. As a result, the DRM is less effective when the right hand side becomes too complicated. In general, it is preferable to keep the right-hand side as simple as possible so that it can be better approximated by the chosen RBFs.

Despite the many special attractive features of RBFs, it is known that most of the RBFs are globally defined basis functions. This means that the resulting matrix for in-

interpolation is dense and can be highly ill-conditioned, especially for a large number of interpolation points in 3D. This poses serious stability problems and high computational cost. Even though domain decomposition offers a remedy for ill-conditioned problems, it requires domain discretization which is a major disadvantage in using the BEM. All of these drawbacks lead us to search for basis functions that have local support. Due to the effort of Schaback and his research group [43, 48, 49], compactly supported positive-definite radial basis functions (CS-RBFs) have been explicitly constructed and applied to multivariate surface reconstruction in mid-1990s. They provide a state-of-the-art interpolation technique and have a firm theoretical basis for large scale interpolation problems. It turns out that the most popular CS-RBFs were the ones constructed by Wendland [48]. Recently, Buhmann [4] has investigated a new, larger class of smooth radial basis functions of compact support which contains other compactly supported ones that were proposed by Wu [49] and Wendland [48]. Due to the virtue of compact support, the implementation of the CS-RBFs leads to a sparse matrix formulation in the approximation of a multivariate function or scattered data. Chen et al [9, 11] implemented Wendland's CS-RBFs [48] in the context of the DRM for solving Poisson's equation in 2D and 3D and later extended it to Helmholtz-type operators in 3D. Fasshauer [15] also implemented these newly constructed CS-RBFs in the context of a Hermite collocation method for solving various kinds of PDEs.

Initially, these new classes of CS-RBFs were considered as a cure for the problems of the dense and ill-conditioned matrices mentioned above. However, several difficulties of the CS-RBFs have been observed: (i) the accuracy and efficiency depends on the scale of the support and 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 become unacceptable. When the support is large enough to make the error acceptable, the matrix system becomes dense and the advantages to the traditional RBFs are lost. As a result, the use of CS-RBFs with a fixed support is not recommended [15]. A multilevel scheme for CS-RBFs was first suggested in the review paper of Schaback [43] and later was fully implemented by Floater and Iske [17] to handle the uncertainty in choosing the size of the support. Based on the multilevel scheme, Fasshauer [15, 16] and Chen et. al [10] also employed multilevel CS-RBFs for solving PDEs.

In this paper we survey recent developments since 1995 of CS-RBFs in the context of the DRM. In Section 2, we briefly introduce the classical RBFs and newly constructed CS-RBFs for scattered data or function interpolation. In Section 3, we employ the method of particular solutions to split the given linear partial differential equation into two parts: homogeneous and nonhomogeneous equations. In Section 4, we introduce the method of fundamental solutions, which requires neither domain nor boundary discretization, to solve the homogeneous equation. In Section 5, we employ the DRM with CS-RBFs as basis functions to approximate the particular solutions. In Section 6, we give a series of known closed-form particular solutions for commonly used differential operators and suggest a new way of deriving analytic particular solutions. In Section 7, a multilevel scheme is introduced to alleviate the difficulty of choosing the scale of support. In Section 8, we discuss iterative methods which do not require the assembling of a matrix. In Section 9, further numerical examples using CS-RBFs have been tested and compared to some other globally supported RBFs. We also found that CS-RBFs are more effective than globally defined RBFs in approximating functions with sharp spikes. In Section 10, methods developed in the previous sections are shown to be ready for implementation for

solving more complicated time-dependent problems which can be achieved by reducing these problems to a series of modified Helmholtz equations using the Laplace transform or time marching schemes. In Section 11, we conclude this article by suggesting a number of topics for future research.

2 Radial Basis Functions

Finite element techniques reconstruct functions from a superpositions of piecewise polynomial functions on subsets of triangulations of a domain or its boundary. In contrast to this, the techniques surveyed here will avoid triangulations and meshing, but they still reconstruct functions by the superposition of simple functions. These simple functions are shifts of radial basis functions (RBFs)

$$\psi(\|x\|_2), \psi : [0, \infty) \rightarrow \mathbb{R} \quad (1)$$

or derivatives thereof. Table 1 provides a selection of cases that can be used without any restrictions. The functions are positive definite (PD) for all space dimensions. In

Gaussian	$\psi(r) = e^{-cr^2}$	$c > 0$
Inverse Multiquadrics	$\psi(r) = (r^2 + c^2)^{\beta/2}$	$c > 0 > \beta$
Sobolev splines	$\psi(r) = K_\nu(r)r^\nu$ $K_\nu =$ spherical Bessel function	$\nu > 0$

Table 1: PD RBFs

certain cases, polynomials up to some small degree $m - 1$ have to be added to ensure a safe reconstruction. More precisely, the functions in question have to be conditionally positive definite (CPD) of some order m , which is zero in the cases of Table 1. Details can be found in [50] or any survey on radial basis functions. We list such functions in Table 2, providing the minimal order m . Note that in these cases the user has to add all polynomials of degree up to $m - 1$ to ensure a safe reconstruction. The situation

Linear	$\psi(r) = r$		$m = 1$
Cubic	$\psi(r) = r^3$		$m = 2$
Polyharmonic splines	$\psi(r) = r^\beta$	$\beta \in \mathbb{R}_{>0} \setminus 2\mathbb{Z}$	$m \geq \lceil \beta/2 \rceil$
Thin-plate splines	$\psi(r) = r^\beta \log r$	$\beta \in 2\mathbb{N}$	$m > \beta/2$
Multiquadrics	$\psi(r) = (r^2 + c^2)^{\beta/2}$	$\beta \in \mathbb{R}_{>0} \setminus 2\mathbb{Z}, c > 0$	$m \geq \lceil \beta/2 \rceil$

Table 2: CPD RBFs

is slightly different for radial basis functions with compact support. In these cases, the order m of positive definiteness is always zero (thus there are no additional polynomials required), but their positive definiteness depends on the space dimension d . In Table 3 we list the compactly supported (unconditionally) positive definite piecewise polynomial radial basis functions (CS-RBFs) of Wendland [48], having minimal degree for prescribed smoothness and space dimension. In Table 3 we have used the cut-off function $(r)_+$ which

$d = 1$	$\psi(r) = (1-r)_+^0$	C^0
	$\psi(r) = (1-r)_+^3 (3r+1)$	C^2
	$\psi(r) = (1-r)_+^5 (8r^2 + 5r + 1)$	C^4
$d = 2, 3$	$\psi_1(r) = (1-r)_+^2$	C^0
	$\psi_2(r) = (1-r)_+^4 (4r+1)$	C^2
	$\psi_3(r) = (1-r)_+^6 (35r^2 + 18r + 3)$	C^4
	$\psi_4(r) = (1-r)_+^8 (32r^3 + 25r^2 + 8r + 1)$	C^6

Table 3: Wendland's CS-RBFs.

is defined to be r if $0 \leq r \leq 1$ and to be zero elsewhere. Furthermore, two examples of the new class of CS-RBFs constructed by Buhmann [4] are also given as follows:

$$\psi(r) = 2r^4 \log r - \frac{7}{2}r^4 + \frac{16}{3}r^3 - 2r^2 + \frac{1}{6}, \quad 0 \leq r \leq 1,$$

$$\psi(r) = \frac{112}{45}r^{9/2} + \frac{16}{3}r^{7/2} - 7r^4 - \frac{14}{15}r^2 + \frac{1}{9}, \quad 0 \leq r \leq 1,$$

and $\phi(r) = 0$ for $r \geq 1$.

For practical implementation, one needs to rescale the support of ψ in (1). This can be achieved by using instead the scaled function

$$\psi^{[\alpha]}(r) = \psi(r/\alpha) \quad (2)$$

for various values of $\alpha > 0$. For a given $x_i \in \Omega$, one defines the scaled CS-RBF ψ_i on Ω as

$$\psi_i^{[\alpha]}(x) = \psi^{[\alpha]}(\|x - x_i\|) \quad x \in \Omega.$$

For the details of the scaling effect, in data fitting, we refer readers to the References [17, 43].

3 The Method of Particular Solutions

First we consider solving the following types of differential equation,

$$Lu(x) = f(x), \quad x \in \Omega, \quad (3)$$

$$u(x) = g_1(x), \quad x \in \Gamma_1, \quad (4)$$

$$\frac{\partial}{\partial n}u(x) = g_2(x), \quad x \in \Gamma_2. \quad (5)$$

where L represents a second order elliptic differential operator with a known fundamental solution, $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a bounded open nonempty domain with sufficiently regular boundary $\partial\Omega = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \cap \Gamma_2 = \emptyset$.

It is well-known that the nonhomogeneous term in (3) can be eliminated by the use of a particular solution. Let

$$u = u_h + u_p \tag{6}$$

where u_p satisfies the nonhomogeneous equation

$$Lu_p(x) = f(x) \tag{7}$$

but does not necessarily satisfy the boundary conditions (4)-(5) and u_h satisfies

$$Lu_h(x) = 0, \quad x \in \Omega, \tag{8}$$

$$u_h(x) = g_1(x) - u_p(x), \quad x \in \Gamma_1, \tag{9}$$

$$\frac{\partial}{\partial n} u_h(x) = g_2(x) - \frac{\partial u_p}{\partial n}, \quad x \in \Gamma_2. \tag{10}$$

Once u_p and $\partial u_p / \partial n$ are known, equations (8)-(10) can be solved by a standard BEM. In the next section, the method of fundamental solutions will be introduced to solve (8)-(10). The final solution of (3)-(5) is then given by $u = u_h + u_p$. The key issue is how to determine the particular solution u_p and its normal derivative.

If $f(x)$ in (7) is simple, u_p may be determined analytically. For general $f(x)$, there are various ways for evaluating particular solutions numerically [2, 18]. After the next section we will focus on how particular solutions can be derived in the context of the DRM using CS-RBFs for various differential operators that are commonly used in the engineering literature.

4 The Method of Fundamental Solutions

Let us assume that the particular solutions, which will be the focus of the next few sections, in (7) have been obtained in some ways. Various boundary methods can be used to approximate the homogeneous solution in (8) - (10). Boundary element methods have been under strong development during the past two decades and are considered as one of the well established numerical techniques for solving PDEs in the areas of science and engineering. However, the BEM also experiences several difficulties in its numerical implementation. The task of surface discretization in 3D is still not trivial and the need to compute tedious near-singular, singular and hyper-singular integrals is tremendous. Moreover, the order of convergence of the BEM is low due to the use of low order polynomial approximations. In recent years, the method of fundamental solutions (MFS) has gained much attention as a boundary-only meshless method. It was originally formulated by Kupradze and Aleksidze [36] and has been further developed by numerous mathematicians and scientists over the past three decades. In the MFS, the singularities are avoided by the use of a fictitious boundary outside the problem domain. The MFS has similar features as the BEM: the need for a fundamental solution and the boundary only formulation. In general, the MFS has the following advantages over its counterpart BEM:

- (i) It requires neither domain nor boundary discretization.
- (ii) No domain integration of any type is required.
- (iii) It converges exponentially for smooth boundary shapes and boundary data.

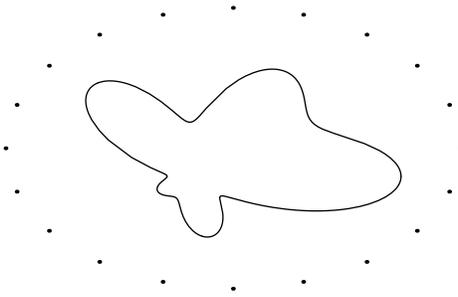


Figure 1: The source points on the fictitious boundary.

- (iv) It is insensitive to the dimensionality of the problem and thus is very attractive for high dimensional problems.
- (v) It is easy for practical implementation and coding.

The MFS is also known as the superposition method [35], desingularized method [5], the charge simulation method [1], or regular BEM in the mathematical and engineering literature. More details about the MFS can be found in two excellent review papers by Fairweather and Karageorghis [14] and Golberg and Chen [21].

Despite the above attractive features, the MFS, however, has never been seriously considered as a major numerical technique in the past partially due to its limitation for solving only homogeneous problems. A key factor that the MFS has gradually revived after three decades of dormancy is that it has been successfully extended to nonhomogeneous problems and various types of time-dependent problems [21, 41] by using the method of particular solutions as discussed in the last section. Most of the important theoretical development of the MFS is also of recent origin [3, 13, 29, 30, 31, 32, 33, 34] and the robustness of the method is greatly enhanced. We refer readers to the review paper by Golberg and Chen [21] for further details on its convergence and stability. Hence, we focus only on the implementation of the MFS in this section.

For convenience, we replace the functions on the right hand sides of (9)-(10) by \tilde{g}_1 and \tilde{g}_2 respectively. In the implementation of the MFS, we assume the approximate solution \tilde{u}_h to the solution of equations (8) - (10) can be expressed as a linear combination of fundamental solutions

$$\tilde{u}_h(x) = \sum_{j=1}^n a_j G(x, y_j), \quad x \in \bar{\Omega}, \quad (11)$$

where $G(x, y_j)$ is the fundamental solution of the linear operator L . Here the singularities, or source points, $\{y_j\}_1^n$ are placed outside the problem domain as shown in Figure 1. Notice that \tilde{u}_h in (11) automatically satisfies the given differential equation (8). All we need to do is to enforce \tilde{u}_h so that it satisfies the boundary conditions (9) - (10). By the collocation method, we choose the same number of collocation points as source points on the physical boundary. In general, we need to add a constant in (11) in the 2D case for completeness purposes [3]. In the real implementation, most of the time we obtained the same solution without adding the extra constant. Hence, we often do not add this extra constant just for convenience in numerical computation. However, we want the readers

to be aware of the existence of such a constant. Let $\{x_i\}_1^{n_1+1} \in \Gamma_1$ and $\{x_i\}_{n_1+2}^{n+1} \in \Gamma_2$, then we have

$$\sum_{j=1}^{n_1} a_j G(x_i, y_j) + a_{n_1+1} = \tilde{g}_1(x_i), \quad 1 \leq x_i \leq n_1 + 1, \quad (12)$$

$$\sum_{j=n_1+1}^n a_j \frac{\partial}{\partial n} G(x_i, y_j) = \tilde{g}_2(x_i), \quad n_1 + 2 \leq x \leq n + 1. \quad (13)$$

The above $(n+1) \times (n+1)$ system of equations can be solved directly by Gaussian elimination. Once all the coefficients $\{a_j\}_1^{n+1}$ are determined, the approximate solution \tilde{u}_h and its derivative $\partial\tilde{u}_h/\partial n$ can be evaluated from (11). As one may have noticed, neither boundary discretization nor boundary integration is required due to the formulation of the MFS in (11). Furthermore, when the boundary geometry, \tilde{g}_1 and \tilde{g}_2 are smooth, \tilde{u}_h often converges exponentially to u_h . As a result, we need only a small number of the collocation points on the physical boundary and fictitious boundary. The given homogeneous equation can be solved much more efficiently than its counterpart BEM. So far as the implementation is concerned, it is straightforward.

Another special feature of the MFS is that by construction, the MFS provides a smooth extension of the solution of the boundary value problem to the exterior of the physical solution domain, even though such an extension may not have a direct physical interpretation. This property is particularly useful when using time-stepping methods for solving time-dependent problems. Due to this property, the accuracy of the MFS does not deteriorate near the boundary as is common in the BEM [20].

Although the MFS is very easy to set up and program as shown above, there are several practical and theoretical issues need to be considered. Two of the most important issues are the choice of the source points $\{y_j\}_1^n$ and collocation points $\{x_j\}_1^{n+1}$. In general, there have been two approaches to choosing $\{y_j\}_1^n$ - fixed and adaptive.

In a fixed method $\{x_j\}_1^{n+1}$ are chosen a priori in some fashion. Much of the work in this direction has relied on the approximation results of Bogomolny [3] and Cheng's convergence results for the Dirichlet problem for Laplace's equation when Ω and the fictitious boundary are concentric circles [13]. In this work it was shown that the accuracy of the approximation improves as the fictitious boundary is moved farther away from $\partial\Omega$. Cheng's result was generalized by Katsurada and Okamoto [29, 30, 31, 32] who showed that if $\partial\Omega$ is a closed Jordan curve in the plane and data are analytic, then

$$\|u_h - \tilde{u}_h\|_\infty \leq c(r/R)^n$$

where r and R are the diameters of D and the fictitious boundary respectively. As a consequence, we found that choosing the source points $\{y_j\}_1^n$ equally spaced around a circle of radius R in \mathbb{R}^2 and equally spaced in polar co-ordinates (ϕ, θ) on a sphere of radius R in \mathbb{R}^3 provided excellent results. In practical cases, because the MFS equations (12) - (13) become highly ill-conditioned as R increases, we have generally limited R to about 5 ~ 10 times the diameter of Ω . Another interesting fact is that despite the ill-conditioning, the accuracy of the numerical solutions is largely unaffected. It is believed that this phenomena can be explained by investigating the singular value decomposition (SVD) of the coefficient matrix of (12) - (13). This work is still under investigation and will be reported in our future work.

In adaptive methods, the coefficients $\{a_j\}_1^n$ and source points $\{y_j\}_1^m$ are determined by satisfying the boundary conditions at $m > n$ boundary points in the least squares sense. This problem is nonlinear in the coordinates of the $\{y_j\}_1^m$ but linear in the coefficients $\{a_j\}_1^n$ and can be time consuming to solve [14]. This approach was first proposed by Mathon and Johnson [38] and further developed by Fairweather and his co-workers and extended to a wide variety of problems; see [14, 26, 27, 28] and references cited therein. However there is little or no theoretical basis for this approach. For convenience, we implement the fixed method for the numerical solution of the homogenous equation in the rest of this paper.

The following fundamental solutions will be frequently used in the following sections. Let $r = \|x - y\|$.

For $L = \Delta$,

$$G(x, y) = \begin{cases} \frac{1}{2\pi} \log(r), & (x, y) \in \mathbb{R}^2, \\ \frac{1}{4\pi r}, & (x, y) \in \mathbb{R}^3, \end{cases}$$

For $L = \Delta - \lambda^2$,

$$G(x, y; \lambda) = \begin{cases} \frac{1}{2\pi} K_0(\lambda r), & (x, y) \in \mathbb{R}^2, \\ \frac{1}{4\pi r} \exp(-\lambda r), & (x, y) \in \mathbb{R}^3, \end{cases}$$

where K_0 is the Bessel function of the third kind with order zero.

Example 1 *To show how the number and location of collocation points and fictitious points affect the final solution in the MFS, we examine a 3D example of the following Laplace equation*

$$\begin{aligned} \Delta u &= 0, & (x, y, z) \in \Omega, \\ u &= e^x \cos y, & (x, y, z) \in \partial\Omega, \end{aligned}$$

where the solution domain Ω is two connected spheres; i.e.,

$$\Omega = \{(x, y, z) \in \mathbb{R}^3 : H(x, y, z) < 1\} \quad (14)$$

with

$$H(x, y, z) = \min \left\{ \left(x - \frac{3}{4} \right)^2, \left(x + \frac{3}{4} \right)^2 \right\} + y^2 + z^2. \quad (15)$$

The profile of Ω , collocation and source points are shown in Figure 2. In general, the fictitious boundary is chosen as a sphere containing Ω . Source points are uniformly distributed on the surface of the fictitious boundary and similarly the collocation points on the surface of the solution domain.

We first fixed the source points on a sphere of radius 9 and various numbers of source and collocation points. Table 4 shows the L_∞ norm at 200 random internal points. Notice that the accuracy improves sharply for the first two hundred points and little improvement afterward.

Next, we fixed the number of source and collocation points and adjust the radius of the fictitious sphere. When n is small, the L_∞ norm deteriorates when the fictitious boundary is close to the solution domain. Table 5 shows the L_∞ norm error with various r and n .

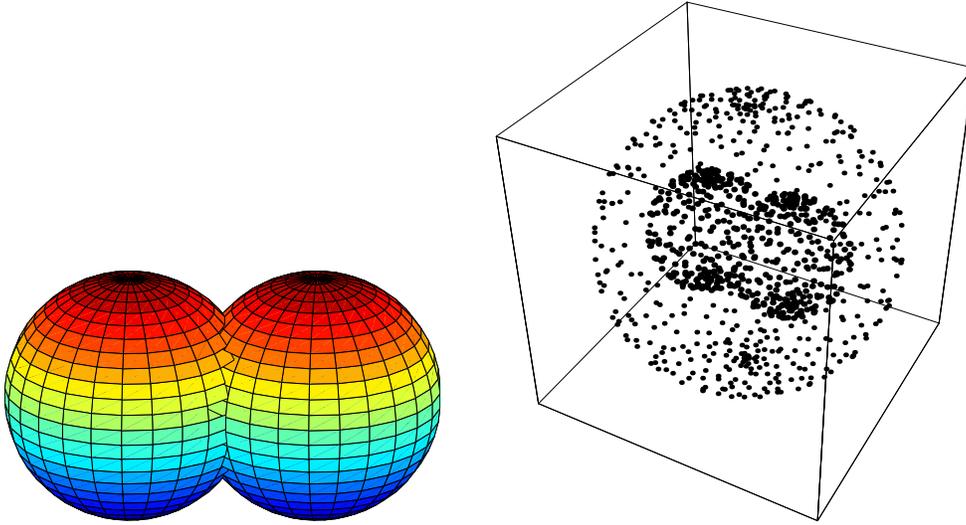


Figure 2: The profiles of solution domain of Ω (left) and scattered points of collocation and source points (right).

n	L_∞ norm	n	L_∞ norm
50	5.58E-2	250	2.32E-09
100	1.41E-2	300	3.71E-09
150	5.88E-5	350	1.73E-10
200	2.43E-9	400	2.02E-10

Table 4: L_∞ norm error with respect to the number of source and collocation points

	$r = 2$	$r = 4$	$r = 6$	$r = 8$	$r = 10$	$r = 12$
$n = 100$	49.363	3.93E-2	7.26E-3	9.51E-3	2.51E-2	8.28E-2
$n = 150$	8.040	1.06E-02	1.15E-3	8.41E-5	6.06E-5	4.36E-5
$n = 200$	2.81E-10	2.01E-10	1.34E-7	2.99E-8	5.89E-9	4.04E-8

Table 5: The effect of the L_∞ norm error with respect to various locations of source points.

5 The Dual Reciprocity Method

The DRM is based on the assumption that one can find an approximation \tilde{f} to the nonhomogeneous term f in (7) and that an analytical solution \tilde{u}_p to

$$L\tilde{u}_p(x) = \tilde{f}(x) \quad (16)$$

can be obtained. Then \tilde{u}_p can be treated as an approximation to a particular solution u_p of (7). The initial step of the DRM is to approximate $f(x)$ by using various kinds of radial basis functions. To avoid the ill-conditioning problem as we indicated in Section 1, we choose CS-RBFs as basis functions. More precisely, we choose a scaling factor α and a set of uniformly distributed points in Ω , say $\{x_j\}_{j=1}^n$ and seek an approximation \tilde{f} to f in the form

$$\tilde{f}(x) := \sum_{j=1}^n a_j \psi_j^{[\alpha]}(x) = \sum_{j=1}^n a_j \psi^{[\alpha]}(\|x - x_j\|), \quad x \in \Omega. \quad (17)$$

The unknown coefficients a_j , $j = 1, \dots, n$ are determined by forcing the interpolatory conditions

$$\tilde{f}(x_i) = f(x_i), \quad 1 \leq i \leq n. \quad (18)$$

Since the CS-RBF $\psi_j^{[\alpha]}$ is positive definite, the solvability of the resulting linear system

$$\sum_{j=1}^n a_j \psi_j^{[\alpha]}(x_i) = f(x_i), \quad 1 \leq i \leq n, \quad (19)$$

is ensured. Let

$$A_{\psi^{[\alpha]}} = \left(\psi^{[\alpha]}(\|x_i - x_j\|) \right)_{i,j=1}^n \quad (20)$$

It is useful to note that $A_{\psi^{[\alpha]}}$ will be dense or sparse depending on the support parameter α .

Once \tilde{f} in (18) has been established, using (16) an approximation \tilde{u}_p (depending on n and α) to a particular solution u_p of (7) can be written as

$$\tilde{u}_p(x) = \sum_{j=1}^n a_j \Psi_j^{[\alpha]}(x), \quad x \in \Omega, \quad (21)$$

where $\Psi_j^{[\alpha]}$ is the solution of

$$L\Psi_j^{[\alpha]}(x) = \psi_j^{[\alpha]}(x), \quad x \in \Omega, \quad j = 1, \dots, n. \quad (22)$$

One of the critical steps in the DRM is the derivation of $\Psi_j^{[\alpha]}$ in (22) analytically. Since the forcing terms in (22) are radially dependent functions on Ω , analytical solvability of (22) for operators L that are radially and translationally invariant can be expected. For such operators if one denotes L_r to be the radial part of L , then finding an analytical representation of \tilde{u}_p in (21) is equivalent to finding the analytical solution $\Psi^{[\alpha]}(r)$ of the radial differential equation

$$L_r \Psi^{[\alpha]}(r) = \psi^{[\alpha]}(r), \quad r \geq 0, \quad (23)$$

where $\psi^{[\alpha]}$ is given by (2) with a typical representation of ψ as in Table 3.

6 Derivation of Analytical Particular Solution $\Psi^{[\alpha]}$

6.1 Laplacian in 2D

In this case, $L_r = (1/r)(d/dr)(rd/dr)$. Explicit analytical representation of $\Psi^{[\alpha]}(r)$ can be derived by straightforward integration; i.e.,

$$\Psi^{[\alpha]}(r) = \begin{cases} \int_0^r \frac{1}{s} \left[\int_0^s t \psi\left(\frac{t}{\alpha}\right) dt \right] ds, & r \leq \alpha, \\ \int_0^\alpha \frac{1}{s} \left[\int_0^s t \psi\left(\frac{t}{\alpha}\right) dt \right] ds + \int_\alpha^r \frac{1}{s} \left[\int_0^\alpha t \psi\left(\frac{t}{\alpha}\right) dt \right] ds, & r > \alpha. \end{cases} \quad (24)$$

The above integration can be performed easily by symbolic software such as MAPLE or MATHEMATICA. A list of particular solutions $\Psi^{[\alpha]}$ corresponding to various Wendland's CS-RBFs is given in the Table 6 (see Reference [9]; note the minor error in the table there). Note that $\{\psi_i\}_{i=1}^4$ are Wendland's CS-RBFs in Table 3.

ψ	Ψ
$\psi_1\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{r^4}{16\alpha^2} - \frac{2r^3}{9\alpha} + \frac{r^2}{4}, & r \leq \alpha, \\ \frac{13\alpha^2}{144} + \frac{\alpha^2}{12} \log\left(\frac{r}{\alpha}\right), & r > \alpha. \end{cases}$
$\psi_2\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{4r^7}{49\alpha^5} - \frac{5r^6}{12\alpha^4} + \frac{4r^5}{5\alpha^3} - \frac{5r^4}{8\alpha^2} + \frac{r^2}{4}, & r \leq \alpha, \\ \frac{529\alpha^2}{5880} + \frac{\alpha^2}{14} \log\left(\frac{r}{\alpha}\right), & r > \alpha. \end{cases}$
$\psi_3\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{7r^{10}}{20\alpha^8} - \frac{64r^9}{27\alpha^7} + \frac{105r^8}{16\alpha^6} - \frac{64r^7}{7\alpha^5} + \frac{35r^6}{6\alpha^4} & r \leq \alpha, \\ -\frac{7r^4}{4\alpha^2} + \frac{3r^2}{4}, & \\ \frac{3517\alpha^2}{15120} + \frac{\alpha^2}{6} \log\left(\frac{r}{\alpha}\right), & r > \alpha. \end{cases}$
$\psi_4\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{32r^{13}}{169\alpha^{11}} - \frac{77r^{12}}{48\alpha^{10}} + \frac{64r^{11}}{11\alpha^9} - \frac{231r^{10}}{20\alpha^8} + \frac{352r^9}{27\alpha^7} & r \leq \alpha, \\ + \frac{352r^9}{27\alpha^7} - \frac{231r^8}{7\alpha^6} + \frac{11r^6}{6\alpha^4} - \frac{11r^4}{16\alpha^2} + \frac{r^2}{4}, & \\ \frac{541961\alpha^2}{8030880} + \frac{7\alpha^2}{156} \log\left(\frac{r}{\alpha}\right), & r > \alpha. \end{cases}$

Table 6: A list of $\Psi^{[\alpha]}$ corresponding to various CS-RBFs $\psi^{[\alpha]}$ in 2D.

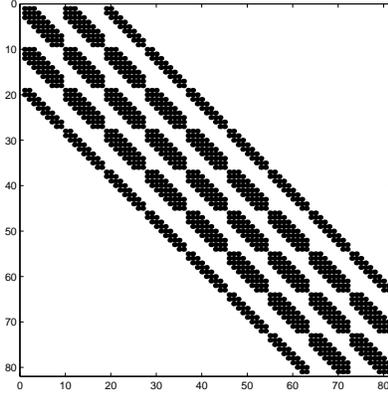


Figure 3: The Profile of the resulting sparse matrix for $\alpha = 0.5$ using CS-RBFs.

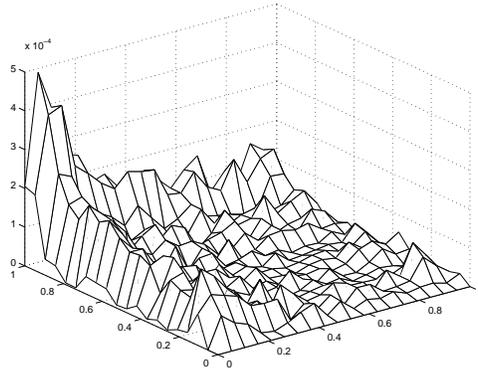


Figure 4: The Profile of the absolute error using ψ_2 with $\alpha = 0.7$.

Example 2 Consider the following problem

$$\Delta u = 2e^{x-y}, \quad (x, y) \in \Omega, \quad (25)$$

$$u = e^{x-y} + e^x \cos y, \quad (x, y) \in \partial\Omega, \quad (26)$$

where $\Omega \cup \partial\Omega$ is the unit square. The exact solution is given by $u = e^{x-y} + e^x \cos y$.

In this example we will show the effect of the scaling factor and smoothness of two different basis functions: $\psi_1 = (1-r)_+^2 \in C^0$. and $\psi_2 = (1-r)_+^4 (4r+1) \in C^2$. Note that 81 evenly distributed points on a regular grid are chosen as the interpolation points in $[0, 1]^2$. L_∞ errors of u are computed on a 20×20 uniform grid in the domain. The numerical tests were performed using MATLAB which has its own sparse matrix solver. The profile of the resulting sparse matrix using a uniform grid for $\alpha = 0.5$ is shown in Figure 3 were used.

The MFS was employed to find the homogeneous solution. 20 uniformly distributed collocation points on the boundary $\partial\Omega$ and 19 uniformly distributed source points on a circle with radius 10 and center $(0, 0)$.

In Table 7, the L_∞ error of u for different scales α and basis functions ψ_1 and ψ_2

	ψ_1	ψ_2		ψ_1	ψ_2
α	L_∞	L_∞	α	L_∞	L_∞
0.3	8.24E-3	5.13E-3	0.9	8.50E-4	3.37E-4
0.4	3.83E-3	1.92E-3	1.0	5.51E-4	3.37E-4
0.5	2.39E-3	1.33E-3	1.1	4.88E-4	3.06E-4
0.6	1.13E-3	5.50E-4	1.2	4.50E-4	2.78E-4
0.7	1.13E-3	4.89E-4	1.3	7.01E-4	2.67E-3
0.8	1.13E-3	3.81E-4	1.4	4.98E-4	1.11E-4

Table 7: L_∞ errors for the approximate solution of u using basis function ψ_1 and ψ_2 .

were computed. The results improve with the increase of the scale of support as one has expected. The larger the scale of the support, the more accurate the interpolation of the forcing term f and u . The accuracy of ψ_2 slightly improves that of ψ_1 due to its higher convergence rate.

6.2 Laplacian in 3D

In this case, $L_r = (1/r^2)(d/dr)(r^2 d/dr)$. Similar to the 2D case, a list of $\Psi^{[\alpha]}$ in Table 8 can be derived [11]. Note that $\{\psi_i\}_{i=1}^4$ are Wendland's CS-RBFs in Table 3.

Example 3 Let us consider the following Poisson problem in 3D:

$$\Delta u = -3 \cos(x) \cos(y) \cos(z), \quad (x, y, z) \in \Omega \quad (27)$$

$$u = \cos(x) \cos(y) \cos(z), \quad (x, y, z) \in \partial\Omega \quad (28)$$

Define $R(\theta) = \sqrt{\cos(2\theta) + \sqrt{1.1 - \sin^2(2\theta)}}$. The surface of the domain $\Omega \cup \partial\Omega$ is represented by the following parametric surface

$$\mathbf{r}(\theta, \phi) = R(\theta) \cos(\theta) \mathbf{i} + R(\theta) \sin(\theta) \cos(\phi) \mathbf{j} + R(\theta) \sin(\theta) \sin(\phi) \mathbf{k}, \quad (29)$$

where $\theta \in [0, \pi)$, $\phi \in [0, 2\pi)$. The analytical solution of (27)-(28) is given by

$$u(x, y, z) = \cos(x) \cos(y) \cos(z) \quad (x, y, z) \in \Omega \cup \partial\Omega. \quad (30)$$

In this example, the basis function $\psi = (1 - r)_+^4 (4r + 1)$ was chosen to approximate the forcing term. 300 quasi-random points [42] were chosen in a box $[-1.5, 1, 5] \times [-.5, .5] \times [.5, .5]$. By the collocation method, particular solutions can be found directly. To solve the sparse system, a real sparse symmetric positive definite linear equation solver (DLSLDX) from the IMSL library (PC version) was employed. In the MFS, 100 quasi-random field points were selected on the parametric surface, as shown in Figure 5, and the same number of quasi-random source points on a sphere with center on origin and radius 10. The numerical results were computed along the x -axis with $y = z = 0$. The results of relative percentage errors with three different scaling factors are shown in Figure 6. Notice that these results are consistent with intuition. With larger support, more interpolation points are included in the process of approximation. Therefore, the more information provided, the more accurate solutions are expected.

ψ	Ψ
$\psi_1\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{r^4}{20\alpha^2} - \frac{r^3}{6\alpha} + \frac{r^2}{6}, & r \leq \alpha, \\ \frac{\alpha^2}{12} - \frac{\alpha^3}{30r}, & r > \alpha. \end{cases}$
$\psi_2\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{r^7}{14\alpha^5} - \frac{5r^6}{14\alpha^4} + \frac{2r^5}{3\alpha^3} - \frac{r^4}{2\alpha^2} - \frac{r^2}{6}, & r \leq \alpha, \\ \frac{\alpha^2}{14} - \frac{\alpha^3}{42r}, & r > \alpha. \end{cases}$
$\psi_3\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{7r^{10}}{22\alpha^8} - \frac{32r^9}{15\alpha^7} + \frac{35r^8}{6\alpha^6} - \frac{8r^7}{\alpha^5} + \frac{5r^6}{\alpha^4} & r \leq \alpha, \\ -\frac{7r^4}{5\alpha^2} + \frac{r^2}{2}, & \\ \frac{\alpha^2}{6} - \frac{8\alpha^3}{165r}, & r > \alpha. \end{cases}$
$\psi_4\left(\frac{r}{\alpha}\right)$	$\begin{cases} \frac{16r^{13}}{91\alpha^{11}} - \frac{77r^{12}}{52\alpha^{10}} + \frac{16r^{11}}{3\alpha^9} - \frac{21r^{10}}{2\alpha^8} + \frac{176r^9}{15\alpha^7} & r \leq \alpha, \\ -\frac{77r^8}{12\alpha^6} + \frac{11r^6}{7\alpha^4} - \frac{11r^4}{20\alpha^2} + \frac{r^2}{6}, & \\ \frac{4903\alpha^2}{60.060} - \frac{8\alpha^3}{165r}, & r > \alpha. \end{cases}$

Table 8: A list of $\Psi^{[\alpha]}$ corresponding to various CS-RBFs $\psi^{[\alpha]}$ in 3D.

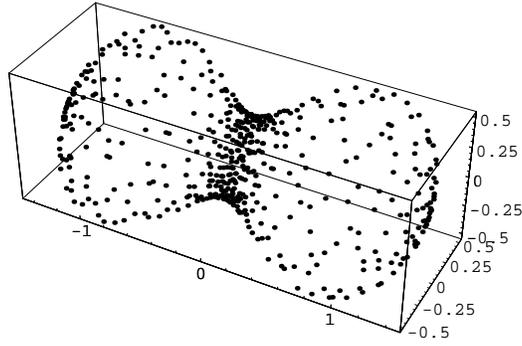


Figure 5: Quasi-random points on the surface as shown in (29).

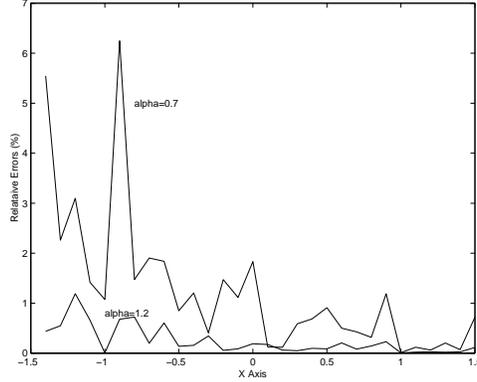


Figure 6: The effect of various scaling factor α

6.3 Helmholtz Equation in 3D

For the modified Helmholtz operator $L = \Delta - \lambda^2$, the derivation of the previous cases by straightforward integration is not feasible. Golberg et al [24] provided an elegant alternative method for deriving $\Psi^{[\alpha]}$ in (22). We briefly summarize it as follows.

Using the radial part of the three dimensional Laplacian, with $L = \Delta - \lambda^2$, equation (22) can be rewritten in the form where p is an appropriately chosen polynomial of degree, say $k \geq 0$, so that the right hand side is a C^{2k} CS-RBF in Table 8.

$$\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\Psi^{[\alpha]}}{dr} \right) - \lambda^2 \Psi^{[\alpha]} = \begin{cases} \left(1 - \frac{r}{\alpha}\right)^n p\left(\frac{r}{\alpha}\right), & 0 \leq r \leq \alpha, \\ 0, & r > \alpha, \end{cases} \quad (31)$$

Note that for $r = 0$, (31) is to be considered in the limiting case as $r \rightarrow 0^+$.

Let

$$\Psi^{[\alpha]}(r) = \frac{w(r)}{r}, \quad r > 0, \quad (32)$$

then one has

$$\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\Psi^{[\alpha]}}{dr} \right) = \frac{1}{r} \frac{d^2 w}{dr^2}.$$

Equation (31) becomes

$$\frac{d^2 w}{dr^2} - \lambda^2 w = \begin{cases} r \left(1 - \frac{r}{\alpha}\right)^n p\left(\frac{r}{\alpha}\right), & 0 \leq r \leq \alpha, \\ 0, & r > \alpha. \end{cases} \quad (33)$$

The general solution of the above equation is given by

$$w(r) = \begin{cases} Ae^{-\lambda r} + Be^{\lambda r} + q(r), & 0 \leq r \leq \alpha, \\ Ce^{-\lambda r} + De^{\lambda r}, & r > \alpha, \end{cases} \quad (34)$$

where $q(r)$ is a polynomial particular solution of the first equation of (33). Note that $q(r)$ can be obtained fairly easily by symbolic ODE solvers such as Maple or Mathematica.

The four coefficients in (34) are to be chosen so that $\Psi^{[\alpha]}$ is twice differentiable at $r = 0, \alpha$ and hence on $[0, \infty)$. For the required differentiability of $\Psi^{[\alpha]}$ at 0, the following theorem [24] shows that it is sufficient to impose the condition that $w(0) = 0$.

Theorem 6.1 *Let w be a solution of (33) with $w(0) = 0$. Then $\Psi^{[\alpha]}$ defined by (32) is twice continuously differentiable at 0 with*

$$\Psi^{[\alpha]}(0) = w'(0), \quad \Psi'^{[\alpha]} = 0, \quad \Psi''^{[\alpha]} = \frac{1}{3} [\lambda^2 w'(0) + p(0)].$$

Furthermore, $\Psi^{[\alpha]}$ satisfies (31) as $\lim_{r \rightarrow 0^+}$.

By the above theorem and (34), the twice continuous differentiability of $\Psi^{[\alpha]}$ at 0 holds if

$$A + B + q(0) = 0. \quad (35)$$

It is easy to show that w is twice differentiable at $r = \alpha > 0$, if

$$\begin{cases} Ae^{-\lambda\alpha} + Be^{\lambda\alpha} = Ce^{-\lambda\alpha} + De^{\lambda\alpha}, \\ -A\lambda e^{-\lambda\alpha} + B\lambda e^{\lambda\alpha} + q'(\alpha) = -C\lambda e^{-\lambda\alpha} + D\lambda e^{\lambda\alpha} \end{cases} \quad (36)$$

In (35) and (36) there are three equations and four unknowns. Hence one of the unknowns can be chosen arbitrary. For convenience, set $D = 0$. Consequently, one has

$$\begin{cases} A = -[B + q(0)], \\ B = -\frac{e^{-\lambda\alpha}[q'(\alpha) + \lambda q(\alpha)]}{2\lambda}, \\ C = B(e^{2\alpha\lambda} - 1) + q(\alpha)e^{\alpha\lambda} - q(0). \end{cases} \quad (37)$$

It follows that one obtains a particular solution

$$\Psi^{[\alpha]}(r) = \begin{cases} \lambda(2B + q(0)) + q'(0), & r = 0, \\ \frac{[Ae^{-\lambda\alpha} + Be^{\lambda\alpha} + q(r)]}{r}, & 0 < r \leq \alpha, \\ \frac{Ce^{-\lambda\alpha}}{r}, & r > \alpha, \end{cases} \quad (38)$$

where A, B and C are as in (37). Notice that for $r > \alpha$ and for $\lambda > 0$ and large, $Ce^{-\lambda\alpha}/r \approx 0$ for $r \gg \alpha$.

In fact, the formula $\Psi^{[\alpha]}(r)$ for the Helmholtz-type operators in (31) holds for all CS-RBFs and only the representation of $q(r)$ in (38) 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 readers to the references [8, 22].

By using symbolic ODE solvers, one has the following:

$$\begin{aligned}
q_1(r) &= \frac{4}{\lambda^4 \alpha} - \left(\frac{1}{\lambda^2} + \frac{6}{\lambda^4 \alpha^2} \right) r + \frac{2}{\lambda^2 \alpha} r^2 - \frac{1}{\lambda^2 \alpha^2} r^3 \\
q_2(r) &= -\frac{480}{\alpha^3 s^6} - \frac{2880}{\alpha^5 s^8} + \left(\frac{1800}{\alpha^4 s^6} + \frac{60}{\alpha^2 s^4} - \frac{1}{s^2} \right) r - \left(\frac{240}{\alpha^3 s^4} + \frac{1440}{\alpha^5 s^6} \right) r^2 \\
&\quad + \left(\frac{300}{\alpha^4 s^4} + \frac{10}{\alpha^2 s^2} \right) r^3 - \left(\frac{20}{\alpha^3 s^2} + \frac{120}{\alpha^5 s^4} \right) r^4 + \frac{15}{s^2 \alpha^4} r^5 - \frac{4}{s^2 \alpha^5} r^6 \\
q_3(r) &= \frac{322\,560}{\lambda^8 \alpha^5} + \frac{7741\,440}{\lambda^{10} \alpha^7} + \left(\frac{168}{\lambda^4 \alpha^2} - \frac{3}{\lambda^2} - \frac{2116\,800}{\lambda^8 \alpha^6} - \frac{12\,700\,800}{\lambda^{10} \alpha^8} - \frac{25\,200}{\lambda^6 \alpha^4} \right) r \\
&\quad + \left(\frac{3870\,720}{\lambda^8 \alpha^7} + \frac{161\,280}{\lambda^6 \alpha^5} \right) r^2 + \left(\frac{28}{\lambda^2 \alpha^2} - \frac{4200}{\lambda^4 \alpha^4} - \frac{2116\,800}{\lambda^8 \alpha^8} - \frac{352\,800}{\lambda^6 \alpha^6} \right) r^3 \\
&\quad + \left(\frac{13\,440}{\lambda^4 \alpha^5} + \frac{322\,560}{\lambda^6 \alpha^7} \right) r^4 - \left(\frac{210}{\lambda^2 \alpha^4} r^5 + \frac{17\,640}{\lambda^4 \alpha^6} + \frac{105\,840}{\lambda^6 \alpha^8} \right) r^5 \\
&\quad + \left(\frac{448}{\lambda^2 \alpha^5} + \frac{10\,752}{\lambda^4 \alpha^7} \right) r^6 - \left(\frac{2520}{\lambda^4 \alpha^8} + \frac{420}{\lambda^2 \alpha^6} \right) r^7 + \frac{192}{\lambda^2 \alpha^7} r^8 - \frac{35}{\lambda^2 \alpha^8} r^9
\end{aligned} \tag{39}$$

where q_1, q_2 and q_3 are $q(r)$ in (38) for $\psi^{[\alpha]} = (1 - r/\alpha)_+^2$, $(1 - r/\alpha)_+^4 (4r/\alpha + 1)$, and $(1 - r/\alpha)_+^6 (35r/\alpha + 18r/\alpha + 3)$ in Table 3 respectively. Note that we have corrected an error of $q_3(r)$ in Example 3 of Reference [24]. The following modified Helmholtz equation was considered by Golberg et al [24].

Example 4 Consider the following Helmholtz problem in 3D

$$(\Delta - 400)u(x, y, z) = -\frac{397}{400}e^{x+y+z}, \quad (x, y, z) \in \Omega, \tag{40}$$

$$u(x, y, z) = \frac{e^{x+y+z}}{400}, \quad (x, y, z) \in \partial\Omega, \tag{41}$$

where the physical domain Ω is two connected spheres in \mathbb{R}^3 which is described in (14)-(15) and Figure 2 in Section 4.

To approximate particular solutions, 400 quasi-random points were generated to serve as the interpolation points in $\Omega \cup \partial\Omega$. The CS-RBF $\psi^{[\alpha]} = (1 - r/\alpha)_+^2$ was chosen as the basis function to interpolate the nonhomogeneous term in (40). The sparseness of the interpolation matrix depends on the scaling factor α . To approximate the homogeneous solution, the MFS with 100 uniformly distributed collocation points on the surface of the physical domain was employed. The same number of source points on the fictitious surface which is a sphere with radius 10 and center $(0,0)$ were chosen.

The L_∞ norm error was computed at 500 random points in Ω for various choices of the scaling factor α . The number of nonzero elements and the sparseness of the interpolation matrix, L_∞ norm, and the corresponding computing time are shown in Table 9. The numerical results shown here are especially encouraging for future work in solving a large class of time-dependent problems.

6.4 Bi-harmonic operator in 3D

The procedure shown above can also be used to find analytic particular solutions for the Laplacian and bi-harmonic operators without integration as shown in (24). Consider the

α	nz	Sparseness(%)	L_∞	CPU (sec.)
0.2	756	0.40	2.28E-2	01.40
0.4	3392	2.10	1.61E-2	04.43
0.6	10562	6.60	9.77E-3	07.48
0.8	21526	13.5	6.45E-3	10.63
1.0	36476	22.8	4.50E-3	13.71
1.2	54116	22.8	3.82E-3	16.98
1.4	73722	46.0	3.30E-3	20.53
1.6	92722	57.9	2.90E-3	24.08
1.8	110498	69.0	2.63E-3	27.60

Table 9: Sparseness, error estimates and CPU time for various α .

bi-harmonic operator $L = \Delta^2$ in 3D. Let

$$W = \frac{1}{r^2} \left(\frac{d}{dr} r^2 \frac{d\Psi^{[\alpha]}}{dr} \right) \quad (42)$$

Then one has

$$\Delta_r^2 \Psi^{[\alpha]} = \frac{1}{r^2} \frac{d}{dr} r^2 \frac{d}{dr} \left[\frac{1}{r^2} \left(\frac{d}{dr} r^2 \frac{d\Psi^{[\alpha]}}{dr} \right) \right] = \frac{1}{r^2} \left(\frac{d}{dr} r^2 \frac{dW}{dr} \right) \quad (43)$$

By a change of variable, one assumes

$$W(r) = \frac{U(r)}{r}. \quad (44)$$

Then,

$$\frac{1}{r^2} \left(\frac{d}{dr} r^2 \frac{dW}{dr} \right) = \frac{1}{r} \frac{d^2 U}{dr^2} = \begin{cases} \left(1 - \frac{r}{\alpha}\right)^n p\left(\frac{r}{\alpha}\right), & 0 \leq r \leq \alpha, \\ 0, & r > \alpha. \end{cases} \quad (45)$$

Equation (45) is a simple differential equation

$$\frac{d^2 U}{dr^2} = \begin{cases} r \left(1 - \frac{r}{\alpha}\right)^n p\left(\frac{r}{\alpha}\right), & 0 \leq r \leq \alpha, \\ 0, & r > \alpha. \end{cases}$$

The general solution of (45) can be found easily. For instance, let $\psi^{[\alpha]} = (1 - \frac{r}{\alpha})_+^4 (4\frac{r}{\alpha} + 1)$, the exact solution U is given by

$$U(r) = \begin{cases} \frac{r^8}{14\alpha^5} - \frac{5r^7}{14\alpha^4} + \frac{2r^6}{3\alpha^3} - \frac{r^5}{2\alpha^2} + \frac{r^3}{6} + A_1 + A_2 r, & 0 \leq r \leq \alpha, \\ C_1 r + C_2, & r > \alpha. \end{cases}$$

where A_1, A_2, C_1 and C_2 are arbitrary constant. Choosing $A_1 = A_2 = 0$,

$$U(r) = \begin{cases} \frac{r^8}{14\alpha^5} - \frac{5r^7}{14\alpha^4} + \frac{2r^6}{3\alpha^3} - \frac{r^5}{2\alpha^2} + \frac{r^3}{6}, & 0 \leq r \leq \alpha, \\ C_1 r + C_2, & r > \alpha. \end{cases}$$

From (44), one has

$$W(r) = U(r)/r = \begin{cases} \frac{r^7}{14\alpha^5} - \frac{5r^6}{14\alpha^4} + \frac{2r^5}{3\alpha^3} - \frac{r^4}{2\alpha^2} + \frac{r^2}{6}, & 0 \leq r \leq \alpha, \\ C_1 + \frac{C_2}{r}, & r > \alpha. \end{cases}$$

From (42),

$$\frac{1}{r^2} \left(\frac{d}{dr} r^2 \frac{d\Psi^{[\alpha]}}{dr} \right) = \begin{cases} \frac{r^7}{14\alpha^5} - \frac{5r^6}{14\alpha^4} + \frac{2r^5}{3\alpha^3} - \frac{r^4}{2\alpha^2} + \frac{r^2}{6}, & 0 \leq r \leq \alpha, \\ C_1 + \frac{C_2}{r}, & r > \alpha. \end{cases}$$

Again, by a change of variable and then repeating the above procedure, one obtains

$$\Psi^{[\alpha]}(r) = \begin{cases} \frac{r^9}{1260\alpha^5} - \frac{5r^8}{1008\alpha^4} + \frac{r^7}{84\alpha^3} - \frac{r^6}{84\alpha^2} + \frac{r^4}{120}, & 0 \leq r \leq \alpha, \\ C_1 r^2 + C_2 r + C_3 \frac{1}{r} + C_4, & r > \alpha. \end{cases}$$

where C_1, C_2, C_3 and C_4 are to be determined by matching the continuity of ψ, ψ', ψ'' and ψ''' at $r = \alpha$. Notice that for $r \leq \alpha$,

$$\begin{aligned} \psi(r) &= \frac{1}{1260} \frac{r^9}{\alpha^5} - \frac{5}{1008} \frac{r^8}{\alpha^4} + \frac{1}{84} \frac{r^7}{\alpha^3} - \frac{1}{84} \frac{r^6}{\alpha^2} + \frac{1}{120} r^4, & \psi(\alpha) &= \frac{\alpha^4}{240} \\ \psi'(r) &= \frac{1}{140} \frac{r^8}{\alpha^5} - \frac{5}{126} \frac{r^7}{\alpha^4} + \frac{1}{12} \frac{r^6}{\alpha^3} - \frac{1}{14} \frac{r^5}{\alpha^2} + \frac{1}{30} r^3, & \psi'(\alpha) &= \frac{4\alpha^3}{315} \\ \psi''(r) &= \frac{2}{35} \frac{r^7}{\alpha^5} - \frac{5}{18} \frac{r^6}{\alpha^4} + \frac{1}{2} \frac{r^5}{\alpha^3} - \frac{5}{14} \frac{r^4}{\alpha^2} + \frac{1}{10} r^2, & \psi''(\alpha) &= \frac{\alpha^2}{45} \\ \psi'''(r) &= \frac{2}{5} \frac{r^6}{\alpha^5} - \frac{5}{3} \frac{r^5}{\alpha^4} + \frac{5}{2} \frac{r^4}{\alpha^3} - \frac{10}{7} \frac{r^3}{\alpha^2} + \frac{1}{5} r, & \psi'''(\alpha) &= \frac{\alpha}{210} \end{aligned} \tag{46}$$

For $r > \alpha$,

$$\psi'(r) = 2C_1 r + C_2 - \frac{C_3}{r^2}, \quad \psi''(r) = 2C_1 + 2\frac{C_3}{r^3}, \quad \psi'''(r) = -6\frac{C_3}{r^4}$$

Hence,

$$\begin{cases} C_1 \alpha^2 + C_2 \alpha + \frac{C_3}{\alpha} + C_4 = \frac{\alpha^4}{240} \\ 2C_1 \alpha + C_2 - \frac{C_3}{\alpha^2} = \frac{4\alpha^3}{315} \\ 2C_1 + \frac{2C_3}{\alpha^3} = \frac{\alpha^2}{45} \\ \frac{-6C_3}{\alpha^4} = \frac{\alpha}{210} \end{cases}$$

Solving the above system of equations, one obtains

$$C_1 = \frac{1}{84}\alpha^2, \quad C_2 = -\frac{1}{84}\alpha^3, \quad C_3 = -\frac{1}{1260}\alpha^5, \quad C_4 = \frac{5}{1008}\alpha^4$$

Unfortunately, the above procedure fails for differential operators in the 2D case. As a result, the closed-form particular solution of Helmholtz-type operators in 2D is still not available.

6.5 Alternative Approach for Analytic Particular Solution

In general, the closed-form particular solutions stated above are difficult to derive. Only for some special differential operators with the properly chosen basis function can the closed form particular solution be obtained. Here we would like to give a simple mathematical statement that will give us alternative to obtain closed-form particular solutions.

If the given differential operator L is positive definite and Ψ is a positive definite radial basis function, then $L\Psi$ is a positive definite function. Using this fact, we can choose a positive definite radial basis function as the particular solution and then produce an interpolation function $\psi = L\Psi$ which is positive definite. In this way, not only is the invertibility of the interpolation matrix guaranteed, but also the particular solution can be obtained automatically. Here we only give preliminary results. Details will be provided in a forthcoming paper.

Example 5 We consider the example of the Poisson problem in (25)-(26). Since CS-RBFs are positive definite functions, we can choose the following basis function as a particular solution

$$\Psi(r) = \left(1 - \frac{r}{\alpha}\right)_+^6 \left(\frac{35r^2}{\alpha^2} + \frac{18r}{\alpha} + 3\right). \quad (47)$$

Then the interpolation function will be

$$\psi(r) = \Delta\Psi(r) = \frac{112}{\alpha^4} \left(1 - \frac{r}{\alpha}\right)_+^4 (20r^2 - 4r\alpha - \alpha^2). \quad (48)$$

The other special feature of this approach is that the particular solution is only evaluated within the cut-off parameter unlike the previous approach where the particular solution at all the interpolation points has to be evaluated. As we notice the overall solution of the earlier approach of computing the particular solution will be more accurate but less efficient for small α as shown in Table 7 and Table 10. However, there is little difference in L_∞ when α becomes large. To perform the numerical computation, we use the same data shown in Example 2 and the results are shown in Table 10. Figure 7 shows the profile of the absolute error using $\alpha = 0.7$.

Example 6 The closed-form particular solution for the Helmholtz equation in 2D is not available in Section 6 due to the difficulty in deriving the particular solution in the reverse order. Since $\Delta - \lambda^2$ is a negative definite differential operator, the above mentioned approach is allowable. This provides a way to produce a particular solution that is not possible in the previous cases. Let us choose the same particular solution Ψ as in the

α	L_∞	α	L_∞
0.3	5.77E-2	0.9	2.47E-4
0.4	2.68E-2	1.0	1.75E-4
0.5	9.10E-3	1.1	2.32E-4
0.6	3.52E-3	1.2	1.28E-4
0.7	1.51E-3	1.3	1.40E-4
0.8	5.68E-4	1.4	1.57E-4

Table 10: L_∞ norm error using new approach.

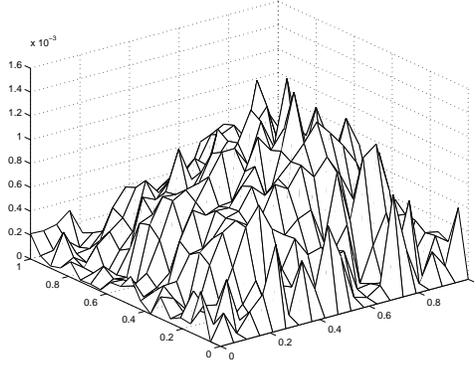


Figure 7: The profile of absolute error using $\alpha = 0.7$.

previous example, then

$$\begin{aligned} \psi(r) &= (\Delta - \lambda^2) \Psi(r) \\ &= \left(1 - \frac{r}{\alpha}\right)_+^4 \left(\frac{112}{\alpha^4} (20r^2 - 4r\alpha - \alpha^2) - \lambda^2 \left(1 - \frac{r}{\alpha}\right)^2 \left(\frac{35r^2}{\alpha^2} + \frac{18r}{\alpha} + 3 \right) \right). \end{aligned}$$

We consider the following Helmholtz equation

$$(\Delta - \lambda^2) u(x, y) = (1 - \lambda^2) (e^x + e^y), \quad (x, y) \in \Omega, \quad (49)$$

$$u(x, y) = e^x + e^y, \quad (x, y) \in \partial\Omega, \quad (50)$$

where Ω is a unit circle. In the numerical computation, we choose $\lambda^2 = 100$ and 150 interpolation points were selected in the domain. For the MFS, 35 evenly distributed collocation points and the same number of source points were chosen. The fictitious boundary is a circle with radius 10 and center at the origin. The L_∞ norm errors are shown in Table 11. Notice that the accuracy continues to improve when the scaling factor increase beyond the full support.

7 Multilevel Schemes for CS-RBFs

As indicated in the Introduction, the determination of the scaling factor α poses a problem. For smaller value of α , the quality of approximation is poor. On the other

α	L_∞	α	L_∞
0.3	2.618	1.1	2.982E-2
0.4	1.711	1.2	2.415E-2
0.5	0.821	1.3	1.978E-2
0.6	0.289	1.4	1.634E-2
0.7	0.098	2.0	6.425E-3
0.8	0.051	4.0	1.228E-3
0.9	4.469E-2	8.0	3.828E-4
1.0	3.658E-2	10	2.965E-4

Table 11: L_∞ norm errors for Helmholtz’s equation.

hand, when α becomes too large, the resulting interpolation matrix using CS-RBFs is no longer sparse and the main purpose of using CS-RBFs is lost. Furthermore, due to the slow convergence rate of CS-RBFs, the determination of both the number of interpolation points and the size of the scaling factor α becomes an important issue. The best way to tackle the above issue is to a certain extent to consider a multilevel scheme by choosing various sizes of interpolation point sets and corresponding cut-off parameters.

Multilevel schemes are a common practice in multivariate data analysis. The basic idea of a multilevel scheme is to capture the main feature of the given function in the first few levels with few interpolation points using large scaling factors and small details are added on in the later steps which consist of a large number of interpolation points but small scaling factors α . Schaback briefly mentioned this in his review paper [43], and later it was fully implemented by Floater and Iske [17] in the context of multivariate interpolation. Chen et al [10] further applied the concept of a multilevel scheme to solving PDEs in the context of the DRM.

Let $D_N = \{x_i\}_{i=1}^N$ be a set of interpolation points. Next one subdivides D_N into a sequence of uniformly distributed point sets

$$D_1 \subset D_2 \subset \dots \subset D_k \subset \dots \subset D_L = D_N$$

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

$$\alpha_1 > \alpha_2 > \dots > \alpha_k > \dots > \alpha_L$$

is a chosen set of cut-off parameters. The choice of α_k and N_k , $k = 1, \dots, l$ depends on the required minimal accuracy of the approximation and the size and sparsity constraints of the interpolation matrix $A_{\psi, \{\alpha_k\}}$.

In [17], a thinning algorithm was devised to produce a sequence of evenly distributed subsets D_k , $k = 1, \dots, l - 1$ of interpolation points. 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. For simplicity, Chen et al [10] used a quasi-Monte Carlo method [42] to generate a sequence of quasi-random points which also ensures that the interpolation points are uniformly distributed at each level. In [10], two multilevel algorithms were proposed to extend the interpolation scheme to solving PDEs. We briefly summarize these approaches in the following subsections.

7.1 Algorithm I for Multilevel CS-RBFs

The first multilevel scheme proposed to find an approximate solution of a PDEs can be described as follows:

For $k = 1, \dots, l$ with α_k being the scaling factor for D_k one sets

$$\tilde{f}^k(x) = \sum_{j=1}^{N_k} c_j^{(k)} \psi_j^{[\alpha_k]} \left(\left\| x - x_j^{(k)} \right\| \right), \quad x_j^{(k)} \in D_k \quad (51)$$

and at level k one chooses the approximate particular solution of (16) as

$$\tilde{u}_p^k(x) = \sum_{j=1}^{N_k} c_j^{(k)} \Psi_j^{[\alpha_k]}(x), \quad x \in \Omega, \quad (52)$$

where $\Psi_j^{[\alpha_k]}$ is a solution of

$$L\Psi_j^{[\alpha_k]}(x) = \psi_j^{[\alpha_k]}(x), \quad x \in \Omega, \quad j = 1, \dots, N_k. \quad (53)$$

The solutions $\Psi_j^{[\alpha_k]}$, $j = 1, \dots, N_k$, $k = 1, \dots, l$ can be computed using analytical formulas such as those described in Section 5. For $k = 1$, the coefficients $c_j^{(1)}$, $j = 1, \dots, N_1$ in (51) and (52) are determined by

$$\tilde{f}^1(x_i) = f(x_i), \quad 1 \leq x_i \leq N_1, \quad (54)$$

and for $k = 2, \dots, l$ the coefficients $c_j^{(k)}$, $j = 1, \dots, N_k$ in (51) and (52) are computed using the interpolatory constraints

$$\tilde{f}^k(x_i) = f(x_i) - \sum_{j=1}^{k-1} \tilde{f}^j(x_i), \quad 1 \leq x_i \leq N_k. \quad (55)$$

Consequently, at each level $k = 1, \dots, l$, the nonhomogeneous function f is approximated by $\sum_{i=1}^k \tilde{f}^i$. Clearly, $f(x) = \lim_{l \rightarrow \infty} \sum_{k=1}^l \tilde{f}^k(x)$, and $\tilde{f}^k(x) \rightarrow 0$ as $k \rightarrow \infty$ for $x \in \Omega$.

At the first level one chooses the support value α_1 high and the number of points N_1 in D_1 low and obtains the unknown coefficient vector $\mathbf{c}^{(1)} = [c_1^{(1)}, \dots, c_{N_1}^{(1)}]^T$ by solving the $N_1 \times N_1$ dense system

$$A_{\psi^{[\alpha_1]}} \mathbf{c}^{(1)} = \mathbf{f}^{(1)} \quad (56)$$

where $\mathbf{f}^{(1)} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_{N_1})]^T$. For subsequent levels $k = 2, \dots, l$, one interpolates the residual of the previous levels. That is for $k = 2, \dots, l$, to compute the vector $\mathbf{c}^{(k)} = [c_1^{(k)}, \dots, c_{N_k}^{(k)}]^T$ one solves the $N_k \times N_k$ finite dimensional systems of the form

$$A_{\psi^{[\alpha_k]}} \mathbf{c}^{(k)} = \mathbf{f}^{(k)} \quad (57)$$

with the first N_{k-1} entries of $\mathbf{f}^{(k)}$ being zeros and the remaining $N_k - N_{k-1}$ entries are given by (55). As the level increases, one decreases the support value and increases the number of interpolation points. Consequently, one solves a sequence of finite dimensional systems with increasing dimension as well as sparsity.

One then continues our algorithm by setting the approximate particular solution \tilde{u}_p^N of (16) as

$$\tilde{u}_p^N = \sum_{k=1}^l \tilde{u}_p^k \quad (58)$$

with \tilde{u}_p^k , $k = 1, \dots, l$ given by (52) and $N = N_l$, the number of chosen interpolation points in the final level.

In general, the particular solution in (16) is not unique and hence the above algorithm yields approximate particular solutions \tilde{u}_p^N converging to a ‘particular’ particular solution of (16). Using the approximate particular solution (58) the final step in our algorithm is to compute an approximate solution u_h^N of the associated approximate homogeneous problem.

One may use boundary integral methods or the MFS to find the homogeneous solution u_h^N . Finally one takes $u^N = u_h^N + \tilde{u}_p^N$ as an approximation to the unique solution u of (3)–(5). The above multilevel algorithm requires a priori the choice of the number of levels l and computation of particular solutions at all the levels $k = 1, \dots, l$ before computing the approximate homogeneous solutions u_h^N . Consequently, in the *algorithm I* one needs a stopping criteria for choosing N depending on TOL .

One approach to tackle this problem is to observe that since $\tilde{f}^k(x) \rightarrow 0$ as $k \rightarrow \infty$ for $\mathbf{x} \in \Omega$, one expects that $\|\mathbf{c}^{(k)}\| \rightarrow 0$ as $k \rightarrow \infty$. Hence, from (52), one expects $\|\tilde{u}_p^k\|_\infty \rightarrow 0$ as $k \rightarrow \infty$. So one may think of using the criteria that one proceed with the above *algorithm I* 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 $\mathbf{c}^{(k)}$ as well as the solutions $\Psi_j^{[\alpha_k]}$ of (53). But for all $j = 1, \dots, N_k$, $\Psi_j^{[\alpha_k]} + \rho$ is also a solution of (53) for any constant ρ where $L(\rho) = 0$. So for example one may have chosen $\Psi_j^{[\alpha_k]} + \frac{1}{TOL}$ (or $\Psi_j^{[\alpha_k]} + \frac{1}{\|\mathbf{c}^{(k)}\|}$) as a solution of (53) there by making the criteria for a $\|\tilde{u}_p^k\|_\infty < TOL$ for a finite k not robust.

A second multilevel algorithm has been proposed to enhance the robustness of the stopping algorithm mentioned above. The trade-off is that some additional computation cost is increased.

7.2 Algorithm II for Multilevel CS-RBFs

Chen et al [10] proposed a multilevel approach by decomposing the exact unique solution of (3)–(5) into a series of solutions of simpler nonhomogeneous equations based on *algorithm I*. Their aim was to devise a multilevel CS-RBF based computational scheme to find approximate solutions of (3)–(5) satisfying a given error tolerance TOL .

Let u_h^1 be the unique solution of

$$Lu_h^1(x) = \tilde{f}^1(x), \quad x \in \Omega, \quad (59)$$

$$u_h^1(x) = g(x), \quad x \in \partial\Omega, \quad (60)$$

and for each $k = 2, 3, \dots$ let v^k be the unique solution of

$$Lv^k(x) = \tilde{f}^k(x), \quad x \in \Omega, \quad (61)$$

$$v^k(x) = 0, \quad x \in \partial\Omega. \quad (62)$$

In (59) and (61), \tilde{f}^k is as defined in (51) with interpolatory constraints given by (54) for $k = 1$ and by (55) for $k = 2, 3, \dots$. Since $f(x) = \sum_{k=1}^{\infty} \tilde{f}^k(x)$, $x \in \Omega$, it is easy to see that the unique solution u of (3)–(5) can be written as

$$u(x) = \sum_{k=1}^{\infty} v^k(x), \quad \mathbf{x} \in \Omega. \quad (63)$$

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

We set an approximate solution of (3)–(5) to be

$$u_N(x) = \sum_{k=1}^l v^k(x), \quad x \in \Omega \quad (64)$$

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 (61)–(62), 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 , $k = 1, \dots, l$, the following procedure was proposed.

For $k = 1$, we write $v^1 = \tilde{u}_p^1 + u_h^1$, where \tilde{u}_p^1 is a particular solution of (59) computed using the representation (52) and (54), and u_h^1 is the approximate solution of the homogeneous problem

$$Lu_h^1(x) = 0, \quad x \in \Omega, \quad (65)$$

$$u_h^1(x) = g(x) - \tilde{u}_p^1(x), \quad x \in \partial\Omega. \quad (66)$$

The homogeneous BVP (3)–(5) can be solved for example using a 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 . Then one computes an approximate solution u_h^1 using the LU factorization with cheaper appropriate matrix-vector multiplications.

For $k = 2, \dots, l$ we write $v^k = \tilde{u}_p^k + u_h^k$, where \tilde{u}_p^k is a particular solution of (61) computed through (52) and (55), and u_h^k is the approximate solution of the homogeneous problem

$$Lu_h^k(x) = 0, \quad x \in \Omega, \quad (67)$$

$$u_h^k(x) = -\tilde{u}_p^k(x), \quad x \in \partial\Omega. \quad (68)$$

Using the LU factorization of M_L , the approximate solution \tilde{u}_p^k at each level $k = 2, \dots, l$ can be computed easily with just appropriate matrix-vector multiplications involving the \tilde{u}_p^k .

It is useful to note that compared to *algorithm I* described earlier, the above algorithm involves solving in addition the homogeneous problems at levels $k = 2, \dots, l$. Since in practice the maximum number of levels l is not expected to exceed 10, the additional matrix-vector multiplications computational cost involved in *algorithm II* is justified

if the 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.

To demonstrate how the multilevel schemes work, we solve the following Poisson problem in the 2D case. We only show the result of algorithm 2 mentioned above. For the details of algorithm 1 and the 3D case, we refer the reader to Reference [10]. Here we reproduce their results with a different solution domain. For the purpose of visual effect, all the figures and the Table below for this test example were obtained on a 25×25 uniform grid in the domain $[1, 2]^2$. Notice that the algorithm works for any arbitrary domain. For each level $k = 1, 2, 3, 4$, we solved the sparse matrix system $A_{\varphi^{[\alpha_k]}} \mathbf{c}^{(k)} = \mathbf{f}^{(k)}$ (see (56) and (57) using the IMSL library sparse matrix solver DLSLDX (PC version). All the numerical computations were performed using double precision. We also choose $TOL = 10^{-3}$.

Example 7 Consider the Poisson problem:

$$\Delta u(x, y) = f(x, y) \quad \text{in } \Omega, \quad (69)$$

$$u(x, y) = g(x, y) \quad \text{on } \partial\Omega. \quad (70)$$

where $\Omega \cup \partial\Omega = [1, 2]^2$. For testing purposes we chose f and g in such a way that the exact solution of (69)–(70) is

$$u(x, y) = \sin \frac{\pi x}{6} \sin \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4}, \quad (x, y) \in \Omega. \quad (71)$$

The choice (71) is possible if the boundary data $g(x, y)$ in (70) is same as in (71) and if the inhomogeneous term $f(x, y)$ is given by

$$\begin{aligned} f(x, y) = & -\frac{751\pi^2}{144} \sin \frac{\pi x}{6} \sin \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4} + \frac{7\pi^2}{12} \cos \frac{\pi x}{6} \cos \frac{7\pi x}{4} \sin \frac{3\pi y}{4} \sin \frac{5\pi y}{4} \\ & + \frac{15\pi^2}{8} \sin \frac{\pi x}{6} \sin \frac{7\pi x}{4} \cos \frac{3\pi y}{4} \cos \frac{5\pi y}{4}. \end{aligned} \quad (72)$$

The profiles of the exact solution (left) and the forcing term $f(x, y)$ (right) are shown in Figure 8. We observe that $f(x, y)$ has a relatively large fluctuation in the domain which makes the approximation more difficult.

To interpolate $f(x, y)$, we choose the CS-RBF $\varphi(r) = (1 - r)_+^4 (4r + 1)$. Using the quasi-Monte Carlo based subroutine SOBSEQ [42], we generated $N = 500$ quasi-random points in the domain. Following our earlier notation, we chose four levels:

$$\alpha_1 = 1.0, \quad \alpha_2 = 0.8, \quad \alpha_3 = 0.5, \quad \alpha_4 = 0.2, \quad \text{and } N_1 = 30, \quad N_2 = 150, \quad N_3 = 300, \quad N_4 = 500$$

and for $k = 1, 2, 3, 4$, D_k consisting of first N_k points from the generated quasi-random points in $[1, 2]^2$.

For $k = 1, 2, 3, 4$ the sparsity structure (with non-zero entries nz_k) of the resulting $N_k \times N_k$ matrix $A_{\varphi^{[\alpha_k]}}$ and the absolute maximum error are given in Table 12.

The profile of the interpolation error of the forcing term $e_k = f - \sum_{i=1}^k f^i$ at each level $k = 1, 2, 3, 4$ are given in Figure 9. The main interest is in the particular solutions obtained at each level. Their profiles are given in Figure 10.

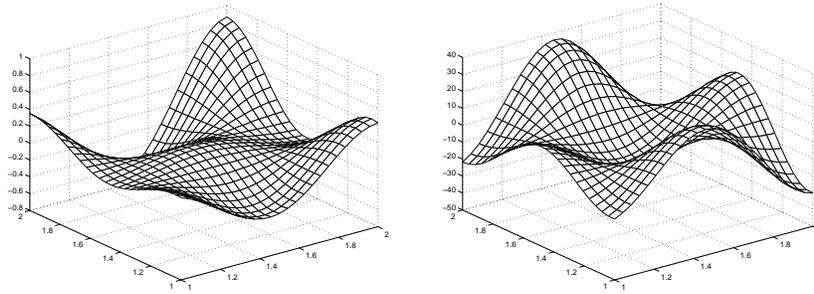


Figure 8: The profiles of exact solution (left) and forcing term $f(x, y)$ (right).

N_k	nz_k	%	L_∞ error
30	459	51	0.1984
150	9729	43.24	0.59E-3
300	22057	24.50	8.86E-3
500	13299	5.31	8.758E-3

Table 12: Sparsity pattern of the interpolation matrix and L_∞ error.

As expected, from Figures 9 and 10, one observes that $\|e_k\|_\infty$ and $\|\tilde{u}_p^k\|_\infty$ get smaller as the level increases; i.e. the scaling factor α shrinks and number of interpolation points increases. The profiles of the solution produced at each level, $\tilde{u}_p^k + u_h^k, k = 1, 2, 3, 4$, are shown in Figure 11. One notes that the contribution of the solution at level 3 and 4 is almost insignificant. The overall errors ‘after’ each level are shown in Figure 12 which are consistent with the solution profile in Figure 11.

8 Iterative Methods for DRBEM with CS-RBFs

For complicated geometrical problems in 3D, large numbers of unknowns are required to represent the solution. The resulting matrix that simulates the problem may be so large that it exceeds the capacity of the computer hardware memory and it becomes impractical to solve it by traditional Gaussian elimination. As a result, the matrix size is often the limiting factor that defines the largest problem a given computer can solve.

In the FDM literature, the so-called relaxation technique [44, 45] has been widely used to circumvent the difficulty of assembling and inverting large scale matrices. The solution process is that an initial trial solution is assigned to a solution grid. The discrete values at each node are corrected in an iterative manner until convergence is achieved. No matrix or matrix elimination process is needed. In order to achieve the goal of devising an efficient algorithm to solve 3D fluid dynamics problems governed by the Navier-Stokes equation, Cheng et al [12] implemented an iterative BEM coupled with the DRM to solve Poisson’s equation. They used the standard DRBEM solution procedure; i.e., the inhomogeneous terms were approximated by a series of radial basis functions. In [12], the iterative method was adopted to evaluate the homogeneous solution using the BEM and inhomogeneous solutions using CS-RBFs which avoid assembling the matrices.

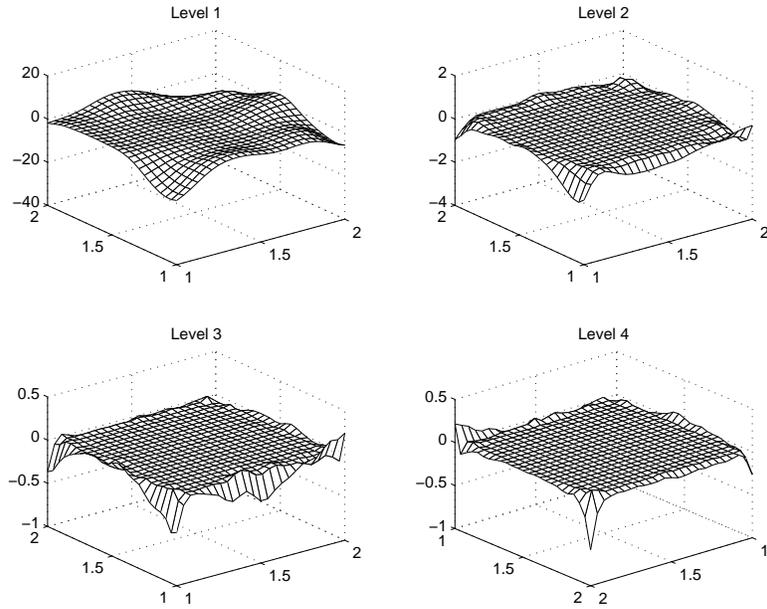


Figure 9: Interpolation errors after each level.

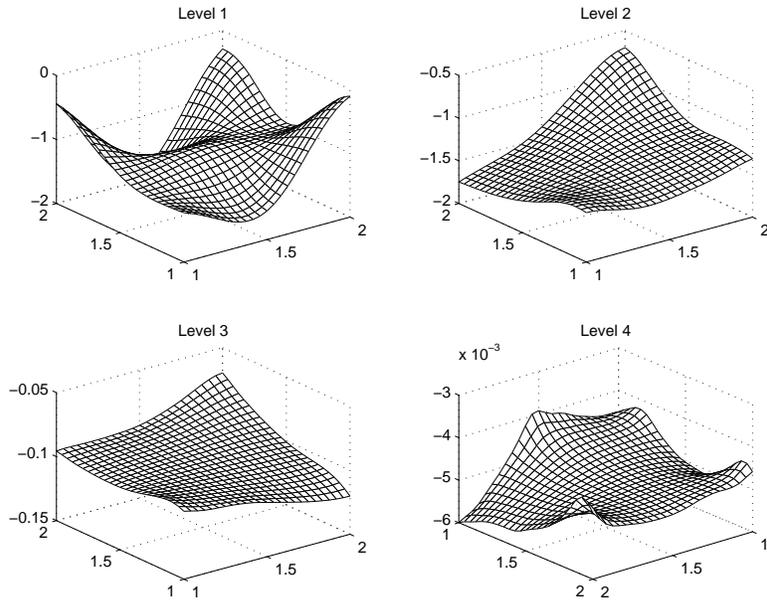


Figure 10: Particular solutions produced at each level.

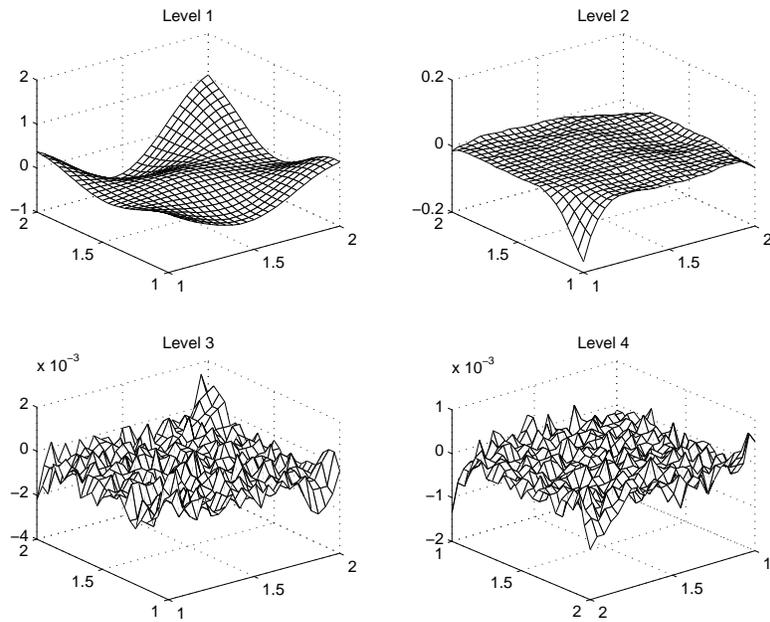


Figure 11: Solution produced at each level.

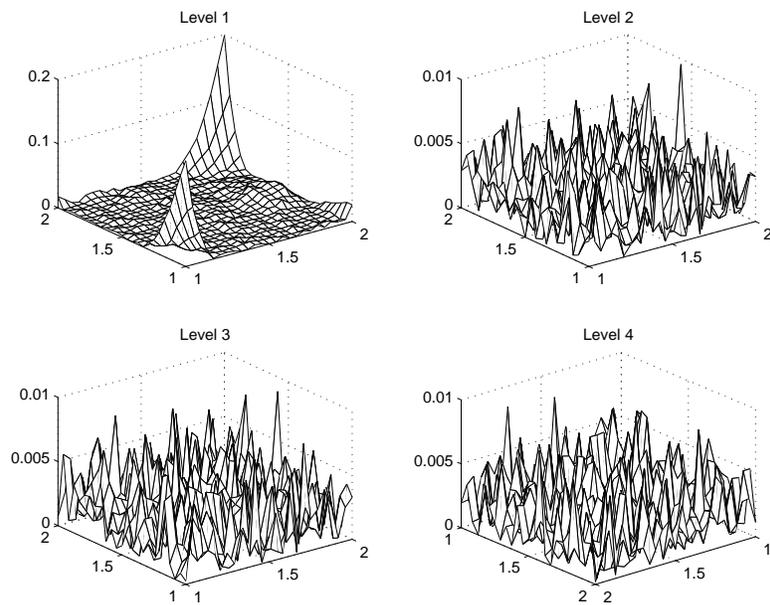


Figure 12: Overall errors produce after each level.

In evaluating the particular solution, the authors claimed that none of the globally supported RBFs converged using iterative methods. Instead, only CS-RBFs achieved convergence. Recently, Tsai et al [51] Young et al [52] extended the same algorithm to solve 3D Poisson's equation and the Stokes flow equation. In fact, we would like to comment that the inverse multiquadric is also a globally supported positive definite RBF. Hence, the iterative method should also work using inverse MQs. We will follow up an example to confirm this shortly.

There are many iterative linear equation solvers available. In [12, 47, 51, 52], two iterative methods, the Gauss-Seidel method and the conjugate gradient method, were presented to solve the homogeneous equation in (8)-(10) and the inhomogeneous equation in (7). Using the method of particular solutions, there are two linear systems to be solved. For the homogeneous equation in (8)-(10), iterative BEMs were implemented. For evaluating the particular solution, first we need to solve the equivalent linear system in (19). Notice that the matrix A in (20) is only imaginary. The detailed implementation of these two iterative schemes can be found in standard textbooks on numerical analysis. Here we present a 2D example for solving Poisson's equation shown in Reference [12]. For the numerical results for the 3D Poisson equation and Stokes flows problems, we refer readers to references [47, 51, 52].

Example 8 *Let us consider the same Poisson problem as in (69)-(70). An 11×11 uniform grid was used for interpolation points. An iterative bi-conjugate gradient method was implemented. In general, an arbitrary initial value can be assigned. In this case, it is noticed that the diagonal terms of the CS-RBF collocation matrix are all equal to unity, one can simply assign the initial trial values as*

$$a_j = f(x_j), \quad i = 1, 2, \dots, n.$$

Two scaling factors, $\alpha = 0.5$ and 1.5 , were chosen for the numerical test. The first order CS-RBF was adopted. For the case of $\alpha = 0.5$, only 8 iterations are needed to converge to a relative tolerance of 10^{-2} , and 17 iterations to a tolerance of 10^{-4} . The iterative scheme is highly efficient. For the case of $\alpha = 1.5$, the number of iterations for convergence is about the same as the case of $\alpha = 0.5$. For the second order CS-RBF, the number of iterations has significantly increased, particularly for large scaling factors, as shown in Figure 3 in Reference [12].

Next, we consider the effect of the number of interpolation points on the convergence rate. The tolerance is set at 10^{-3} . The number of iterations versus the number of collocation points were shown in Figure 4 in Reference [12]. The result indicated a relation

$$\text{iteration number} \sim (\text{number of interpolation points})^{1/2}$$

The accuracy of the approximation was also examined. The maximum relative error using the 11×11 grid and the first order CS-RBF with $\alpha = 0.5$ is 2.8%. For the second order CS-RBF, the accuracy is only slightly improved, with a relative maximum error of -2.5% . Little improvement was reported for the larger scaling factors. Based on these tests shown in Reference [12], it was concluded that the first order CS-RBF with $\alpha = 0.5$ should be used to proceed with solving the homogeneous equation.

For comparison, other globally supported RBFs were also tested. It was observed that the thin plate spline gave a relative maximum error of 1.7% and the first order conical RBF 2.6%. Note that in both of these two cases, the iterative scheme failed and matrix elimination methods were used.

CS-RBF			Inverse MQ		
α	L_∞ norm	# iteration	c	L_∞ norm	# iteration
0.5	1.368E-3	30	0.6	8.071E-5	157
0.7	4.894E-4	36	0.8	6.528E-6	197
1.0	2.239E-4	42	1.0	7.632E-5	198
1.2	1.842E-4	45	1.5	3.658E-6	109
1.4	1.811E-4	47	2.0	1.412E-6	156

Table 13: Comparison of iterate method using CS-RBF and inverse MQ.

Example 9 To show the difference of iterative methods using CS-RBFs and inverse MQ, we redo the example of the 2D Poisson equation in (25)-(26). We choose the CS-RBF $\psi^{[\alpha]} = (1 - r/\alpha)_+^4(4r/\alpha + 1)$ and the inverse MQ $\psi = 1/\sqrt{(r^2 + c^2)}$, where c is the shape parameter, and the particular solution corresponding to Ψ is given by $\Psi = \sqrt{r^2 + c^2} - c \log(c + \sqrt{r^2 + c^2})$. The tolerance for the convergence of the iteration is fixed at 10^{-5} . The L_∞ norm error and the number of iterations required are shown in the following table. These results were obtained using the MATLAB iterative solver BICG using the bi-conjugate gradient method. In the table, we note that, using CS-RBFs, the overall accuracy and the number of iterations depend on the scale of the support which contradicts [12]. The inverse MQ produces higher accuracy but at the cost of efficiency; i.e., higher number of iterations. However, we also found that if the tolerance for the convergence of the iteration was reduced to 10^{-3} , and the overall accuracy is in the range of 10^{-4} and the number of iterations is significantly reduced to the range between 30 and 40.

Finally, we would like to comment that there are positive definite RBFs other than CS-RBFs that can be constructed.

9 Numerical Comparison on CS-RBFs and Other RBFs

In this section we examine the performance of the CS-RBFs and compare it to MQ and thin plate splines. We also notice that the condition numbers of CS-RBFs are quite high when the scaling factor becomes large. We consider the following benchmark problem [37]:

$$\Delta u(x, y) = -\frac{5\pi^2}{4} \sin \pi x \cos \frac{\pi y}{2}, \quad \text{in } \Omega, \quad (73)$$

$$u(x, y) = \sin \pi x \cos \frac{\pi y}{2}, \quad \text{on } \partial\Omega. \quad (74)$$

where $\Omega = [0, 1]^2 \setminus [0.5, 1]^2$ which is three-quarters of the unit square. The exact solution is given by $u(x, y) = \sin \pi x \cos \frac{\pi y}{2}$.

Here we choose 81 evenly distributed collocation points in $\bar{\Omega}$. For the basis functions, we choose the 3rd order CS-RBFs with scaling factor $\alpha = 1.2$, the MQ with shape parameter $c = 0.8$ and the 3rd order polyharmonic spline $r^6 \log r$. In the MFS, we use 16 evenly distributed points on the boundary $\partial\Omega$ and same number of source points on

a circle with center at $(0.5, 0.5)$ and radius 5. In the following figures, the contour plots of the solution u and velocity field ∇u are plotted on the left and the contour plot of the solution and the distribution of absolute errors are plotted on the right. As far as the accuracy is concerned, MQ and 3rd order polyharmonic splines are comparable. As expected, CS-RBFs can't compete with global functions in accuracy. However, the CS-RBFs provide the flexibility of choosing the local support and thus offer the possibility of solving large scale problems.

The condition number of the 3rd order CS-RBF, MQ and 3rd order polyharmonic splines are 9.76×10^5 , 1.33×10^7 and 6.38×10^7 .

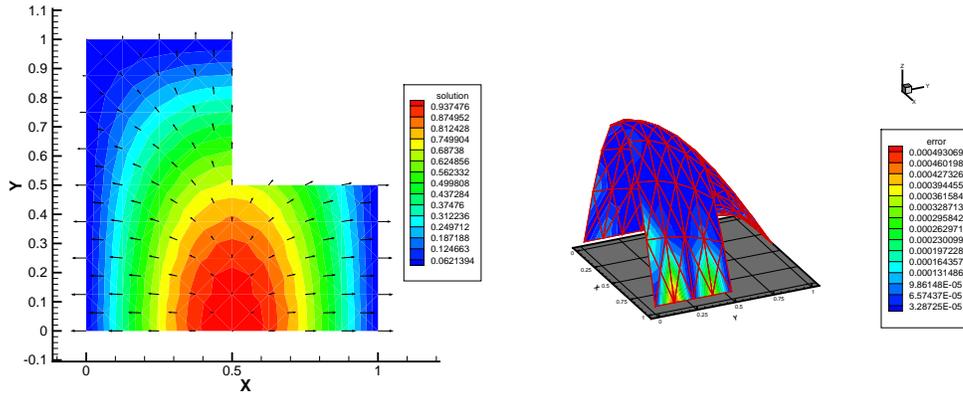


Figure 13: The 3rd order CS-RBFs with $\alpha = 1.2$: contour plot of the solution and velocity field (Left) and contour plot of the solution and error distribution (Right).

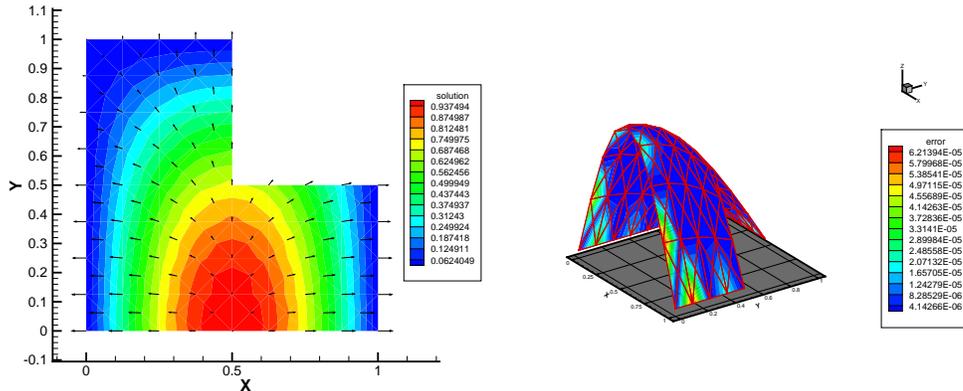


Figure 14: MQ with $c = 0.8$: contour plot of the solution and velocity field (Left) and contour plot of the solution and error distribution (Right).

In general, the globally defined RBFs are better than CS-RBFs in terms of accuracy. For CS-RBFs, as shown in Sections 6.1 and 6.2, we also demonstrated that larger support normally resulted in better accuracy using CS-RBFs. We will show that the above perception is not necessarily true at least for the following type of problem which contains sharp spikes in the right-hand side.

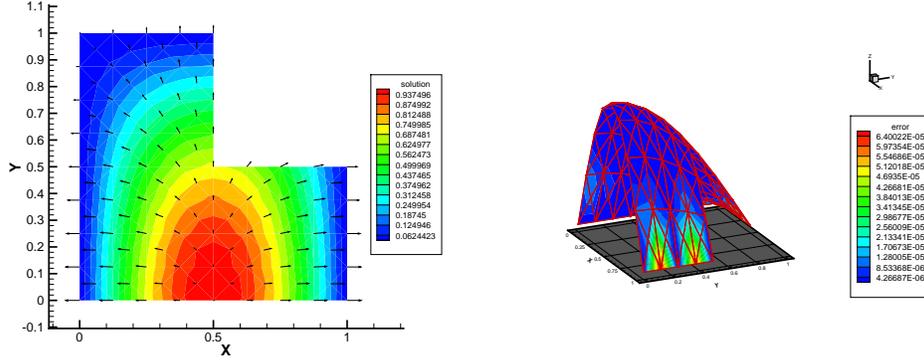


Figure 15: The 3rd order polyharmonic splines: contour plot of the solution and velocity field (Left) and contour plot of the solution and error distribution (Right).

We consider the following Poisson problem

$$\Delta u = \frac{-4a^2 + 3a\sqrt{x^2 + y^2} - (x^2 + y^2)}{(\sqrt{x^2 + y^2} - a)^3}, \quad (x, y) \in \Omega, \quad (75)$$

$$u = \frac{(x^2 + y^2)}{a - \sqrt{x^2 + y^2}}, \quad (x, y) \in \partial\Omega, \quad (76)$$

where $\Omega \cup \partial\Omega = [0, 1]^2$. The exact solution of (75)-(76) is given by

$$u(x, y) = \frac{(x^2 + y^2)}{a - \sqrt{x^2 + y^2}}. \quad (77)$$

The right-hand side of (75) presents a singularity at the corner $(1, 1)$ when $a = \sqrt{2}$. To demonstrate the effectiveness of the CS-RBFs in approximating sharp spike functions, we chose to test the cases $a = 1.5$ and 1.6 . The profiles of the right-hand side of (75) and the exact solution for $a = 1.5$ are shown in Figure 16 in which the range of the right-hand side is $[8/3, 7343]$. For $a = 1.6$, the range of the right-hand side is $[8/3, 850]$. The stiffness for the case of $a = 1.6$ greatly declines but a sharp spike is still present. It is obvious that the problem is much easier to handle for the case $a = 1.6$ than $a = 1.5$.

First, we chose 400 uniformly distributed random points to interpolate the right-hand side and evaluated the L_∞ norm at 400 even grid points. We chose $\varphi = (1 - r)_+^4 (4r + 1)$ as the basis function to approximate the right-hand side. In Table 14, we observe that for $a = 1.5$ the results are much less satisfactory than for $a = 1.6$. Next, we use the 400 regular grid points as interpolation points and compute the L_∞ norm at 400 random points. In Table 15, we see that there is an improvement in accuracy for the case $a = 1.6$, but no effect for the case $a = 1.5$. Furthermore, one also sees that increasing the support α does not really improve the accuracy as we may have expected.

One quick way to improve the performance for the case $a = 1.5$ is to redistribute the interpolation points. Since we know that the spike is known to exist near $(1, 1)$ and the absolute minimum occurs at $(0, 0)$, we propose to enhance the interpolation points in the regions near these two points as shown in Figure 17. Note that one should

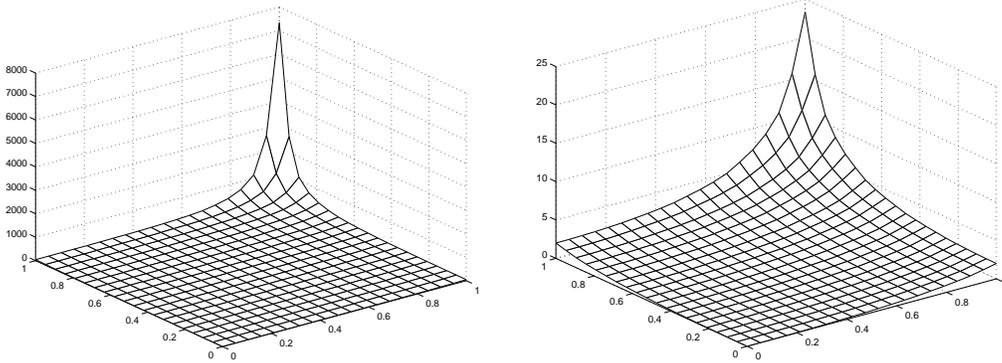


Figure 16: The profiles of the forcing term (left) and exact solution (right).

α	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$L_\infty(a = 1.5)$	0.227	0.267	0.236	0.236	0.236	0.167	0.205
$L_\infty(a = 1.6)$	0.0313	0.0220	0.0125	0.0116	0.0160	0.0121	0.00816

Table 14: L_∞ norm errors using random interpolation points.

be cautious in deploying these points. Too many points deployed in a small region may cause serious ill-conditioning problem, while few additional enhanced points will not have enough effect to capture the special feature of the spike and thus be less effective. For Test 1, 60 additional points were distributed in each of the enhanced regions $[0, 0.3]^2$ and $[0.8, 1.0]^2$. For Test 2, only 30 additional points each were distributed in smaller regions $[0, 0.3]^2$ and $[0.9, 1.0]^2$. In Table 16, we show the numerical results of these tests. The overall profile of absolute maximum errors for Test 2 with $\alpha = 0.5$ is shown in Figure 18. Comparing results in Table 14 and Table 16, it is clear that the accuracy can be improved by simply redistributing the interpolation points in a certain fashion. With little effort, the improvement of the accuracy is about one order of magnitude. In Table 16, we also observe that the increase of the support does not necessarily improve the accuracy for $\alpha > 0.5$.

We also remark that we have tested several globally defined RBFs for this problem and all of them produced unacceptable errors. When the sharp spike in $f(x, y)$ is even larger such as the case for $a = 1.46$ ($f(1, 1) = 45133$), the effort of just enhancing interpolation points in particular regions as discussed above is not enough. We are currently investigating other near singular nonhomogeneous problems by removing the sharp spikes before we use CS-RBFs to approximate the right-hand side. We will present

α	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$L_\infty(a = 1.5)$	0.594	0.417	0.250	0.167	0.241	0.236	0.166
$L_\infty(a = 1.6)$	0.0335	0.0155	0.0161	0.00624	0.00515	0.00580	0.00575

Table 15: L_∞ norm errors using regular grid points.

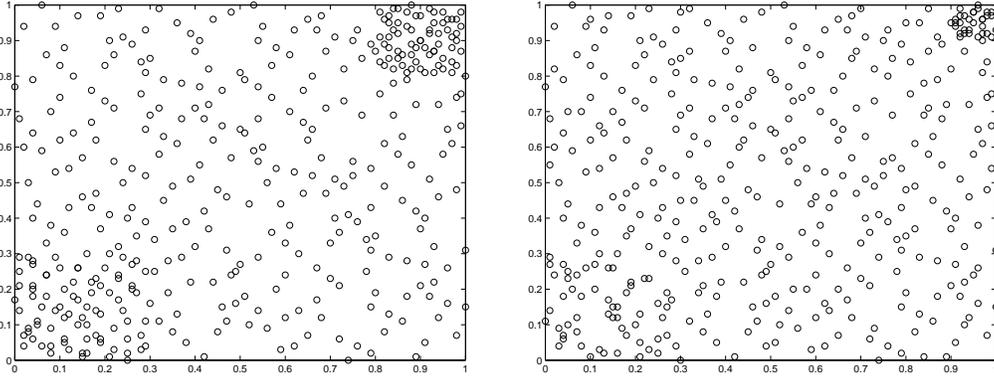


Figure 17: The profiles of distributing points in Test 1 (left) and in Test 2 (right).

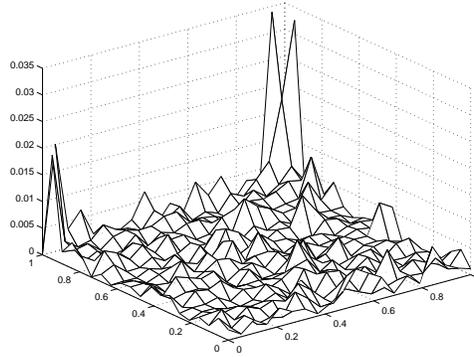


Figure 18: Profile of overall L_∞ norm errors with $a = 1.5, \alpha = 0.5$.

our results in forthcoming papers.

10 Time-Dependent Problems

In this section we show how a number of second order time-dependent partial differential equations can be solved numerically by reformulating them in terms of a sequence of inhomogeneous modified Helmholtz equations. Recently, due to the availability of the analytic particular solution of the modified Helmholtz equation using polyharmonic splines, such an approach has been implemented in the context of the DRM for effectively solving time-dependent problems [25, 39]. In the past, the analytic particular solutions for Helmholtz-type equations using CS-RBFs were only available for 3D case and have never been implemented in time-dependent problems. Using our new approach for evaluating particular solutions given in Section 6.5, it becomes possible to solve time-dependent problems using CS-RBFs in 2D. We will briefly outline the solution process and report our preliminary results in the following subsections.

α	$L_\infty(a = 1.5)$		$L_\infty(a = 1.6)$	
	Test 1	Test 2	Test 1	Test 2
0.2	$2.56E - 1$	$1.42E - 1$	$4.52E - 2$	$1.13E - 1$
0.3	$1.26E - 1$	$5.56E - 2$	$1.15E - 2$	$1.01E - 2$
0.4	$4.46E - 2$	$3.39E - 2$	$4.93E - 3$	$1.01E - 2$
0.5	$2.93E - 2$	$5.27E - 2$	$2.68E - 3$	$7.50E - 3$
0.6	$1.42E - 1$	$9.14E - 2$	$6.09E - 3$	$2.15E - 3$
0.7	$6.28E - 2$	$8.72E - 2$	$4.89E - 3$	$2.44E - 3$
0.8	$8.09E - 2$	$8.42E - 2$	$3.53E - 3$	$1.91E - 3$

Table 16: L_∞ norm errors for Test 1 and Test 2.

10.1 The diffusion equation

Let us first consider boundary value problems for the diffusion equation

$$\Delta u(x, t) - u_t(x, t) = f(x, t), \quad x \in \Omega \subseteq \mathbb{R}^d, \quad d = 2, 3, \quad (78)$$

where Ω is a bounded domain in \mathbb{R}^d , $d = 2, 3$. Without loss of generality, we assume Dirichlet boundary conditions

$$u(x, t) = g(x, t), \quad x \in \partial\Omega, \quad t > 0 \quad (79)$$

and the initial condition

$$u(x, 0) = h(x), \quad x \in \Omega. \quad (80)$$

To solve (78)-(80) numerically, two approaches are considered: (i) taking the Laplace transforms in 't' [8, 53] and (ii) finite differencing in 't' [6]. Defining the Laplace transform

$$U(x, s) = \int_0^\infty e^{-st} u(x, t) dt, \quad s > M \quad (81)$$

and applying (81) to (78)-(80) U satisfies the boundary value problem

$$\Delta U(x, s) - sU(x, s) = \tilde{f}(x, s) - h(x) \equiv m(x, s), \quad x \in \Omega, \quad (82)$$

$$U(x, s) = \tilde{g}(x, s), \quad x \in \partial\Omega, \quad (83)$$

where \tilde{f} and \tilde{g} are Laplace transform of f and g respectively. If $m(x, s) \neq 0$, then (82) is an inhomogeneous modified Helmholtz equation and can be solved using the algorithm in Section 6.5.

Because numerical inversion of the Laplace transform is an ill-posed problem, the above algorithm may not be effective for all problems. Alternatively, one can proceed by using finite difference schemes in time. A popular class of methods for doing this are the θ -methods, defined as follows [6]: let $\tau > 0$ and define the mesh $t_n = n\tau$, $n \geq 0$. For $t_n \leq t \leq t_{n+1}$, approximate $u(x, t)$ by ($0 \leq \theta \leq 1$)

$$u(x, t) \simeq \theta u(x, t_{n+1}) + (1 - \theta)u(x, t_n) \quad (84)$$

so that

$$\Delta u(x, t) \simeq \theta \Delta u(x, t_{n+1}) + (1 - \theta) \Delta u(x, t_n) \quad (85)$$

and

$$u_t(x, t) \simeq \frac{u(x, t_{n+1}) - u(x, t_n)}{\tau}. \quad (86)$$

substituting (85)-(86) in (78) and denoting the resulting approximation to $u(x, t_n) \equiv u_n(x)$ by $v_n(x)$, one obtains

$$\theta \Delta v_{n+1} + (1 - \theta) \Delta v_n - \frac{(v_{n+1} - v_n)}{\tau} = f_n \quad (87)$$

so that ($f_n \equiv f(x, t_n)$)

$$\Delta v_{n+1} - \frac{v_{n+1}}{\theta \tau} = \frac{-v_n}{\theta \tau} - \frac{(1 - \theta) \Delta v_n}{\theta} + \frac{f_n}{\theta}. \quad (88)$$

For $\theta = 1$, we get the backward difference scheme [6]

$$\Delta v_{n+1} - \frac{v_{n+1}}{\tau} = \frac{-v_n}{\tau} + f_n \quad (89)$$

and for $\theta = 1/2$, we get the Crank-Nicholson scheme

$$\Delta v_{n+1} - \frac{2v_{n+1}}{\tau} = \frac{-2v_n}{\tau} - \Delta v_n + 2f_n. \quad (90)$$

One may now observe that (88) is a sequence of inhomogeneous modified Helmholtz equations which can be solved using $v_0 = h$ and the boundary conditions $v_n(x) = g(x, t_n), x \in \partial\Omega$. As before, v_{n+1} can be determined using the MFS once the right hand side of (88) is known. Particular solutions can then be determined using CS-RBFs which is discussed in detail in Section 6.

10.2 The wave equation

Similarly, the wave equation can be reformulated in a similar fashion as the heat equation. Here, we consider the IBVP

$$\Delta u(x, t) = u_{tt}(x, t), \quad x \in \Omega, \quad t > 0, \quad (91)$$

$$u(x, 0) = u_0, \quad u_t(x, 0) = v_0, \quad (92)$$

with Dirichlet boundary conditions

$$u(0, t) = g(x, t), \quad x \in \partial\Omega, \quad t > 0. \quad (93)$$

If one takes the Laplace transform of (91), then the Laplace transform U of u satisfies

$$\Delta U(x, s) = sU(x, s) - su_0 - v_0, \quad x \in \Omega, \quad (94)$$

$$U(x, s) = \tilde{g}(x, s) \quad x \in \partial\Omega. \quad (95)$$

From (94) we see that U satisfies an inhomogeneous modified Helmholtz equation so the determination of U proceeds as for the diffusion equation. To obtain an approximation in 't' one needs to invert the numerical approximation to U .

To avoid transform inversion problems one can resort to time-differencing, as for the diffusion equation. Generalizing the approach of Su and Tabarrok [46], one can define

a class of θ algorithms as follows: approximate u_{tt} by the central difference formula ($u_n \equiv u(x, t_n)$)

$$u_{tt} \simeq \frac{u_{n+1} - 2u_n + u_{n-1}}{\tau^2} \quad (96)$$

and

$$(\Delta u)_n \simeq \theta (\Delta u)_{n+1} + (1 - \theta) (\Delta u)_{n-1}. \quad (97)$$

Substituting (96) and (97) into (91) and denoting the resulting approximation to u_n by v_n , it satisfies

$$\theta \Delta v_{n+1} - \frac{v_{n+1}}{\tau^2} = \frac{v_{n-1} - 2v_n}{\tau^2} + (1 - \theta) \Delta v_{n-1}. \quad (98)$$

For $\theta = 1/2$ we get a second order accurate Crank-Nicholson scheme [46].

Again we see for $\theta \neq 0$ that v_n satisfies a sequence of inhomogeneous Helmholtz equations which can be solved in the same fashion as for the diffusion equation.

Other types of time-dependent problems can also be reformulated in a similar fashion as the previous two cases. The research in time-dependent problems using CS-RBFs in the context of the DRM is still not available.

Example 10 Consider the heat equations in (78)-(80) where $f(x, y, t) = 0$ and $g(x, y, t)$ and $h(x, y)$ are chosen in a way such that the exact solution $u(x, t)$ is given by

$$u(x, y, t) = \frac{1}{4\pi(t + 0.1)} \exp\left(-\frac{(x^2 + y^2)}{4(t + 0.1)}\right). \quad (98)$$

The solution domain Ω is an ellipse with major axis 3 and minor axis 1. For the time difference scheme, we chose $\theta = 1$ in (88) and $\tau = 0.01$. We also choose the particular solution and interpolation function as shown in Section 6.5 for modified Helmholtz equation. We used a fixed scaling factor $\alpha = 1.5$. In the interpolation of the inhomogeneous term at each time step, we generated 300 quasi-random points as we have done previously. For the MFS, 40 collocation points were evenly placed on the boundary $\partial\Omega$ and an equal number of source points are placed on the fictitious boundary of a circle with radius 9 and center $(0, 0)$. At each time step, the solution u at each interpolation point has to be evaluated, then they are used to interpolate the right hand side of (89). The following table shows the solution at $(0, 0)$.

Time	Exact Sol.	Abs. Error	Time	Exact Sol.	Abs. Error
0.2	0.2653	0.2939E-02	1.2	0.06121	0.3466E-02
0.4	0.1592	0.3567E-02	1.4	0.05305	0.2617E-02
0.6	0.1137	0.4653E-02	1.6	0.04681	0.1801E-02
0.8	0.08842	0.4762E-02	1.8	0.04188	0.1060E-02
1.0	0.07234	0.4249E-02	2.0	0.03789	0.4038E-03

Table 17: Maximum absolute errors at various time steps at point $(0, 0)$.

11 Conclusions

In this article, we have surveyed the research work on CS-RBFs since they first appeared in the BEM literature and made a number of proposals and new directions for future research. Coupled with the MFS, we have actually achieved a meshless method for solving partial differential equations. The methodology described in this article provides a solid mathematical foundation and is especially attractive for solving higher dimensional, complicated geometrical and large scale problems efficiently. The method is ready to take on the challenge for solving more applied problems such as the Navier-Stokes equation, moving boundary value problems, etc. Furthermore, we also provide some preliminary new results of our current research work. We encourage interested readers to further investigate and develop practical numerical schemes for solving large scale industry problems.

Acknowledgment: The first author acknowledges the support of a Faculty Development Leave from University of Nevada, Las Vegas, U.S.A., and the support of host institutions at the University of Göttingen and the University of Erlangen-Nürnberg, Germany, during Spring 2001.

References

- [1] K. Amano. A charge simulation method for the numerical conformal mapping of interior, exterior and doubly connected domains. *Journal of Computational and Applied Mathematics*, 53:353–370, 1994.
- [2] K. E. Atkinson. The numerical evaluation of particular solutions for Poisson’s equation. *IMA Journal of Numerical Analysis*, 5:319–338, 1985.
- [3] A. Bogomolny. Fundamental solutions method for elliptic boundary value problems. *SIAM J. Numer. Anal.*, 22:644–669, 1985.
- [4] M.D. Buhmann. A new class of radial basis functions with compact support. *Mathematics of Computation*, 70:307–318, 2001.
- [5] Y. Cao, W. Schultz, and R. Beck. Three dimensional desingularized boundary integral methods for potential problems. *Int. J. Num. Meth. Eng.*, 12:785–803, 1991.
- [6] R. Chapko and R. Kress. Rothe’s method for the heat equation and boundary integral equations. *J. of Integral Equations and Applications*, 9:47–68, 1997.
- [7] C. S. Chen and Y. F. Rashed. Evaluation of thin plate spline based particular solutions for Helmholtz-type operators for the DRM. *Mech. Res. Comm.*, 25:195–201, 1998.
- [8] C. S. Chen, Y. F. Rashed, and M. A. Golberg. A mesh-free method for linear diffusion equations. *Numer. Heat Transf., Pt. B*, 33:469–486, 1989.
- [9] C.S. Chen, C.A. Brebbia, and H. Power. Dual reciprocity method using compactly supported radial basis functions. *Comm. Num. Meth. Eng.*, 15:137–150, 1999.

- [10] C.S. Chen, M.A. Golberg, M. Ganesh, and A. H.-D. Cheng. Multilevel compact radial functions based computational schemes for some elliptic problems. to appear in *Computers and Mathematics with Application*.
- [11] C.S. Chen, M. Marcozzi, and S. Choi. The method of fundamental solutions and compactly supported radial basis functions - a meshless approach in 3D problems. In C.A. Brebbia and H. Power, editors, *Boundary Element Methods XXI*, pages 313–322. WIT Press, 1999.
- [12] A.H.-D. Cheng, D.-L. Young, and J.-J. Tsai. Solution of Poisson’s equation by iterative DRBEM using compactly supported, positive definite radial basis function. *Eng. Anal. Boundary Elements*, 24:549–557, 2000.
- [13] R.S.C. Cheng. *Delta-Trigonometric and Spline Methods using the single-layer Potential representation*. PhD thesis, University of Maryland, 1987.
- [14] G Fairweather and A. Karageorghis. The method of fundamental solution for elliptic boundary value problems. *Advances in Computational Mathematics*, 9:69–95, 1998.
- [15] G. Fasshauer. Solving differential equations with radial basis functions: multilevel methods and smoothing. *Adv. in Comp. Math.*, 11:139–159, 1998.
- [16] G. Fasshauer, C. Gartland, and J. Jerome. Algorithms defined by Nash iteration: some implementations via multilevel collocation and smoothing. *J. Comp. Appl. Math.*, 119:161–183, 2000.
- [17] M. S. Floater and Iske A. Multistep scattered data interpolation using compactly supported radial basis functions. *J. Comp. Appl. Math.*, 73:65–78, 1996.
- [18] M.A. Golberg. The numerical evaluation of particular solutions in the BEM - a review. *Boundary Elements Comm.*, 6:99–106, 1995.
- [19] M.A. Golberg and C.S. Chen. The theory of radial basis functions applied to the BEM for inhomogeneous partial differential equations. *Boundary Elements Communications*, 5:57–61, 1994.
- [20] M.A. Golberg and C.S. Chen. *Discrete Projection Methods for Integral Equations*. Comp. Mech. Publ., 1997.
- [21] M.A. Golberg and C.S. Chen. The method of fundametal solutions for potential, Helmholtz and diffusion problems. In M.A. Golberg, editor, *Boundary Integral Methods: Numerical and Mathematical Aspects*, pages 103–176. WIT Press, 1998.
- [22] M.A. Golberg and C.S. Chen. An efficient mesh free method for solving nonlinear diffusion-reaction equations. *Computer Modeling in Engineering & Sciences*, 1:81–88, 2001.
- [23] M.A. Golberg, C.S. Chen, H. Bowman, and H. Power. Some comments on the use of radial basis functions in the dual receprocity method. *Computational Mechanics*, 21:141–148, 1998.
- [24] M.A. Golberg, C.S. Chen, and M. Ganesh. Paticular solutions of 3D Helmholtz-type equations using compactly supported radial basis functions. *Eng. Anal. with Boundary Elements*, 24:539–547, 2000.

- [25] M.A. Golberg, C.S. Chen, and A.S. Muleshkov. The method of fundamental solutions for time-dependent problems. In C.A. Brebbia C.S. Chen and D.W. Pepper, editors, *Boundary Element Technology XIII*, pages 377–386. WIT Press, 1999.
- [26] A. Karageorghis. Modified methods of fundamental solutions for harmonic and biharmonic problems with boundary singularities. *Numerical Methods for Partial Differential Equations*, 8:1–19, 1992.
- [27] A. Karageorghis and G. Fairweather. The method of fundamental solutions for the numerical solution of the biharmonic equation. *Journal of Computational Physics*, 69:435–459, 1987.
- [28] A. Karageorghis and G. Fairweather. The method of fundamental solutions for the solution of nonlinear plane potential problems. *IMA Journal of Numerical Analysis*, 9:231–242, 1989.
- [29] M. Katsurada. Asymptotic error analysis of the charge simulation method in a jordan region with an analytic boundary. *Journal of the Faculty of Science of Tokyo University, Section 1A*, 37:635–657, 1990.
- [30] M. Katsurada. Charge simulation method using exterior mapping functions. *Japan Journal of Industrial and Applied Mathematics*, 11:47–61, 1994.
- [31] M. Katsurada and H. Okamoto. A mathematical study of the charge simulation method. *Journal of the Faculty of Science, University of Tokyo, Section 1A*, 35:507–518, 1988.
- [32] M. Katsurada and H. Okamoto. The collocation points of the fundamental solution method for the potential problem. *Computers and Mathematics with Applications*, 31:123–137, 1996.
- [33] T. Kitagawa. On the numerical stability of the method of fundamental solutions applied to the Dirichlet problem. *Japan Journal of Applied Mathematics*, 35:507–518, 1988.
- [34] T. Kitagawa. Asymptotic stability of the fundamental solution method. *Journal of Computational and Applied Mathematics*, 38:263–269, 1991.
- [35] G.H. Koopman, L. Song, and J.B. Fahnlne. A method for computing acoustic fields based on the principle of wave superposition. *Journal of Acoustical Society of American*, 86:2433–2438, 1988.
- [36] V.D. Kupradze and M.A. Aleksidze. A method for the approximate solution of limiting problems in mathematical physics. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 4:199–205, 1964.
- [37] J. Li and C.S. Chen. On the numerical tests using compactly supported radial basis functions in the DRM. in preparation.
- [38] R. Mathon and R. L. Johnston. The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM J. Numer. Anal.*, 14:638–650, 1977.

- [39] A.S. Muleshkov, M.A. Golberg, and C.S. Chen. Particular solutions of Helmholtz-type operators using higher order polyharmonic splines. *Comp. Mech.*, 23:411–419, 1999.
- [40] D. Nardini and C. A. Brebbia. A new approach to free vibration analysis using boundary elements. In C.A. Brebbia, editor, *Boundary Element Methods in Engineering*, pages 312–326. Springer-Verlag, 1982.
- [41] P.W. Partridge, C.A. Brebbia, and L.C. Wrobel. *The Dual Reciprocity Boundary Element Method*. CMP/Elsevier, 1992.
- [42] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in Fortran: the Art of Scientific Computing*. Cambridge University Press, 1996.
- [43] R. Schaback. Creating surfaces from scattered data using radial basis functions. In T. Lyche M. Dæhlen and L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, pages 477–496. Vanderbilt University Press, 1995.
- [44] R.V. Southwell. *Relaxation methods in engineering science, A treatise on approximate computation*. Oxford University Press, 1940.
- [45] R.V. Southwell. *Relaxation methods in theoretical physics*. Oxford University Press, 1946.
- [46] J. Su and B. Tabarrok. A time-marching integral equations method for unsteady state problems. *Computer Methods in Applied Mechanics and Engineering*, 142:203–214, 1997.
- [47] C.C. Tsai, D.L. Young, and A.H.-D. Cheng. Meshless BEM for steady three-dimensional Stokes flows. to appear.
- [48] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.
- [49] Z. Wu. Multivariate compactly supported positive definite radial functions. *Advances in Computational Mathematics*, 4:283–292, 1995.
- [50] Z. Wu and R. Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA Journal of Numerical Analysis*, 13:13–27, 1993.
- [51] D.L. Young, C.C. Tsai, and A.H.-D. Cheng. An iterative DRBEM for three-dimensional Poisson's equation. In C.A. Brebbia, editor, *Boundary Element Technology XIV*, pages 323–332. WIT Press, 2001.
- [52] D.L. Young, C.C. Tsai, T.I. Eldho, and A.H.-D. Cheng. Solution of Stokes flow using an iterative DRBEM based on compactly-supported, positive-definite radial basis function. to appear in *International Journal of Computer and Mathematics with Applications*.
- [53] S. Zhu, P. Satravaha, and X. Lu. Solving linear diffusion equations with the Dual Reciprocity Method in Laplace space. *Eng. Anal. with Boundary Elements*, 13:1–10, 1994.