Computers and Mathematics with Applications 71 (2016) 171–184

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

An improved localized method of approximate particular solutions for solving elliptic PDEs

Guangming Yao

Department of Mathematics, Clarkson University, Potsdam, NY 13699-5815, United States

ARTICLE INFO

Article history:
Received 15 July 2015
Received in revised form 28 October 2015
Accepted 9 November 2015
Available online 27 November 2015

Keywords:
Radial basis function
Elliptic equation
Polyharmonic splines
Gaussian
Matérn
MQ

ABSTRACT

In this paper we improve the localized method of approximate particular solutions (LMAPS) in Yao et al. (2011) by utilizing the polyharmonic splines (PS) radial basis function (RBF) for solving elliptic partial differential equations (PDEs). LMAPS has been widely circulated since it is published in 2010. The multiquadric (MQ) has been considered as the most popular choice among all RBFs. However, adjusting the shape parameter is a critical issue when utilizing the original LMAPS. In this paper, we modified LMAPS by combining conditionally positive definite RBF-PS and an additional low degree of polynomial basis in the localization process. The accuracy of the proposed LMAPS is significantly improved. We can simply increase the order of PS to achieve even higher accuracy. Other than the unexpected high accuracy, there is no need to deal with the difficult issues of choosing optimal shape parameter. This is a huge advantage in the RBF simulations of PDEs. In the numerical experiments, we will present the pros and cons of improved LMAPS (ILMAPS) using PS and some commonly used RBFs (MQ, Matérn, and Gaussian) versus the original LMAPS (OLMAPS).

1. Introduction

A global version of the method of approximate particular solution (MAPS) has been developed to solve various types of partial differential equations (PDEs) [1,2]. Similar to the indirect radial basis function (RBF) collocation method (IRBFCM) introduced in [3–5], this numerical scheme interpolates the forcing term, and then the interpolation is integrated and collocated on the domain of interest. Unlike IRBFCM which integrates RBFs in the Cartesian coordinate system, MAPS integrates RBFs in the radial coordinate system, which constitutes a very simple and very effective scheme. In MAPS, the particular solutions of a differential operator with typical RBFs in the right-hand side leads to the particular solutions depending on the radius only, i.e., they are radial basis functions too. As shown by many other researchers, the use of particular solutions is slightly more accurate compared with the use of other typical RBFs directly [6,4].

In [7,8], we proposed a localized meshless methods, the localized method of particular solutions (LMAPS), which can be applied to many problems in science and engineering [9–12]. A main feature of LMAPS is that collocation by MAPS takes place on overlapping local domains. This drastically reduces the size of the collocation matrix. The price paid, however, is that many small matrices must now be solved. Additionally, a global sparse system has to be constructed and to be solved for the approximations. It is apparent that LMAPS can compete with traditional numerical methods for solving large-scale PDEs. Numerical integration is not needed in LMAPS, which makes it easy for this approach to handle extremely irregular domains.

E-mail address: gyao@clarkson.edu.

http://dx.doi.org/10.1016/j.camwa.2015.11.008
0898-1221/

In [8], the multiquadric (MQ) and inverse multiquadric (IMQ) are used in LMAPS. Thereafter, Chen [13,11] et al. extended the method to Gaussian and Matérn RBFs. The particular solutions of the Laplace and the modified Helmholtz operators with multiple RBFs such as MQ, Matérn, and Gaussian in the right-hand side are available in the literature [13,11,8]. These are positive definite RBFs [14–16]. In this paper, we improve LMAPS so that conditional positive definite RBF-polyharmonic splines (PS) can be also used for solving elliptic PDEs using LMAPS. This is not a trivial extension as the invertibility of the small matrices generated during the localization process cannot be achieved unless a low-order polynomial basis is added to the PS kernels [17–19]. The method becomes extremely accurate without much computational cost added.

In Section 2, we propose the formulation of the improved LMAPS utilizing PS on a general elliptic boundary value problem. In Section 3, we list the RBFs considered, and corresponding particular solutions (see Appendix). In Section 4, three elliptic PDEs are considered in irregular domains. We will illustrate performance of our improved method through numerical experiments on a Poisson equation, a modified Helmholtz equation, and eventually a general variable coefficient second-order elliptic PDE in irregular domains. A comparison of improved LMAPS using PS, MQ, Gaussian, and Matérn RBFs with the original LMAPS is done on these three examples, in terms of accuracy, efficiency, and stability. In Section 5, we draw some conclusions on the use of PS in LMAPS compared to other RBFs.

2. Improved LMAPS for elliptic PDEs using polyharmonic splines

In this section we present the improved LMAPS for solving elliptic PDEs of the form

$$\mathscr{L}u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{1}$$

$$\mathscr{B}u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \tag{2}$$

where $\mathscr{L}$ and $\mathscr{B}$ are linear partial differential operators, $\Omega \subset \mathbb{R}^2$ is a bounded and closed domain with boundary $\partial\Omega$. We propose to revise LMAPS using PS in the two-dimensional case. The improved LMAPS can be extended by replacing PS by other RBFs.

The PS RBFs in $\mathbf{R}^2$ are defined as:

$$\phi(r) = r^{2m}\ln(r),$$

where $m \in \mathbb{N}$, in which PS is a general case of the thin plate splines (TPS): $\phi(r) = r^2\ln(r)$. Franke [20] and Buhmann [21] demonstrated the importance of polyharmonic splines in even-dimensional spaces. The interpolation matrix generated by PS can be singular, even with non-trivial sets of distinct centers [22,23]. Typically, to insure the invertibility of the interpolation system, a low-order polynomial basis has to be added to the RBF interpolant. Furthermore, the PS RBFs is conditionally positive definite. Let $\mathscr{P}$ be a polynomial function space of dimension q, where

$$q = \frac{(m+2)(m+1)}{2}.$$

Let $\{p_1, p_2, \ldots, p_q\}$ be a basis of $\mathscr{P}$. Let $\{\mathbf{p}_j\}_{j=1}^N \in \Omega \cup \partial\Omega$ be given interpolation points, where the first N_I interior points are in Ω, followed by N_B boundary points on $\partial\Omega$, and $N = N_I + N_B$. If $\{f(\mathbf{p}_j)\}_{j=1}^{N_I}$ and $\{g(\mathbf{p}_j)\}_{j=N_I+1}^{N}$ are given, LMAPS can approximate the solution to (1)–(2) at a set of evaluation points. For simplicity, we consider the approximation of u at given interpolation node $\mathbf{p}_j$, $j = 1, 2, \ldots, N$, and denote the approximate values by $\hat{u}(\mathbf{p}_j), j = 1, 2, \ldots, N$.

For any point $\mathbf{p} \in \Omega \cup \partial\Omega$, we create an influence domain $\Omega_{\mathbf{p}}$, which contains the n nearest interpolation points to $\mathbf{p}$, where $n \ll N$. For convenience, we denote that the local domain of influence at interpolation point $\mathbf{p}_j$ by $\Omega_j = \{\mathbf{p}_k, k = 1, 2, \ldots, n\}$. Note these points are associated with the point $\mathbf{p}_j$. To discretize (1)–(2) at every $\mathbf{p}_j$, we apply MAPS on every local domain Ω_j. To overcome the difficulty of singularity from utilizing particular solutions of PS, we add an additional polynomial basis to the particular solution kernels. The solution to (1)–(2) can be approximated by particular solutions (which are RBFs too) and the augmented polynomials in the following manner:

$$u(\mathbf{p}) \approx \hat{u}(\mathbf{p}) = \sum_{i=1}^{n} \alpha_i \Phi(\|\mathbf{p} - \mathbf{p}_i\|) + \sum_{l=1}^{q} \alpha_{n+l} p_l(\mathbf{p}), \quad \mathbf{p} \in \Omega_j \tag{3}$$

with additional polynomial basis $p_l(\mathbf{p}) = x^l y^{q-l}$, $l = 1, 2, \ldots, q$, and

$$\mathscr{L}\Phi(r) = \phi(r), \tag{4}$$

where $\phi(r) = r^{2m}\ln(r)$. Since there are q additional degrees of freedoms in (3), the standard polynomial insolvency constraint [3] must be applied. Thus the collocation technique gives the following linear system:

$$\sum_{i=1}^{n} \alpha_i \Phi(\|\mathbf{p}_k - \mathbf{p}_i\|) + \sum_{l=1}^{q} \alpha_{n+l} p_l(\mathbf{p}_k) = \hat{u}(\mathbf{p}_k), \quad k = 1, 2, \ldots, n, \tag{5}$$

$$\sum_{i=1}^{n} \alpha_i p_l(\mathbf{p}_i) = 0, \quad l = 1, 2, \ldots, q. \tag{6}$$

Note that (5)–(6) is a linear system of equations with $n+q$ coefficients which are to be determined. Denote the coefficient matrix in the first term in (5) as $\mathbf{\Phi}_{nn}$, and the second term as $\mathbf{P}_{nq}$. Note the sub-indices indicate the size of the vector or the matrices. Then the system (5)–(6) can be rewritten in block matrix form

$$\begin{bmatrix} \mathbf{\Phi}_{nn} & \mathbf{P}_{nq} \\ \mathbf{P}_{nq}^T & \mathbf{0}_{qq} \end{bmatrix} \boldsymbol{\alpha}_{n+q} = \begin{bmatrix} \hat{\mathbf{u}}_n \\ \mathbf{0}_q \end{bmatrix}, \tag{7}$$

where $\hat{\mathbf{u}}_n = [\hat{u}(\mathbf{p}_1), \hat{u}(\mathbf{p}_2), \ldots, \hat{u}(\mathbf{p}_n)]^T$, $\boldsymbol{\alpha}_{n+q} = [\alpha_1, \alpha_2, \ldots, \alpha_{n+q}]^T$.

Denote the coefficient matrix in (7) as $\mathbf{\Phi}_{n+q}$. It can be proved that $\mathbf{\Phi}_{n+q}$ is non-singular such that the inverse matrix can always be computed provided that all of the nodal points inside Ω_j are distinct points. The unknown coefficients in (5)–(6) can be written as follows

$$\boldsymbol{\alpha}_{n+q} = \mathbf{\Phi}_{n+q}^{-1} \begin{bmatrix} \hat{\mathbf{u}}_n \\ \mathbf{0}_q \end{bmatrix}. \tag{8}$$

Hence, for any $\mathbf{p} \in \Omega_j$, $\hat{u}(\mathbf{p})$ can be expressed in terms of the function values at n nodal points, $\hat{\mathbf{u}}_n$, i.e.

$$\begin{aligned} \hat{u}(\mathbf{p}) &= \sum_{i=1}^{n} \alpha_i \Phi(\|\mathbf{p} - \mathbf{p}_i\|) + \sum_{l=1}^{q} \alpha_{n+l} p_l(\mathbf{p}) \\ &= \mathbf{\Phi}_{n+q}(\mathbf{p}) \boldsymbol{\alpha}_{n+q} \\ &= \mathbf{\Phi}_{n+q}(\mathbf{p}) \mathbf{\Phi}_{n+q}^{-1} \begin{bmatrix} \hat{\mathbf{u}}_n \\ \mathbf{0}_q \end{bmatrix} \\ &= \mathbf{\Psi}_{n+q}(\mathbf{p}) \begin{bmatrix} \hat{\mathbf{u}}_n \\ \mathbf{0}_q \end{bmatrix} \\ &= \mathbf{\Psi}_n(\mathbf{p}) \hat{\mathbf{u}}_n, \end{aligned} \tag{9}$$

where

$$\mathbf{\Phi}_{n+q}(\mathbf{p}) = \left[\Phi(\|\mathbf{p} - \mathbf{p}_1\|), \ldots, \Phi(\|\mathbf{p} - \mathbf{p}_n\|), 1, x, y, x^2, xy, y^2, \ldots, x^q, x^{q-1}y, \ldots, xy^{q-1}, y^q\right],$$

and

$$\mathbf{\Psi}_{n+q}(\mathbf{p}) = \mathbf{\Phi}_{n+q}(\mathbf{p}) \mathbf{\Phi}_{n+q}^{-1}, \tag{10}$$

with $\mathbf{\Psi}_n(\mathbf{p})$ is a sub-vector whose components are the first n elements of $\mathbf{\Psi}_{n+q}(\mathbf{p})$.

Let

$$\hat{\mathbf{u}} = [\hat{u}(\mathbf{p}_1), \hat{u}(\mathbf{p}_2), \ldots, \hat{u}(\mathbf{p}_N)]^T. \tag{11}$$

We will reformulate (9) in terms of global $\hat{\mathbf{u}}$ instead of local $\hat{\mathbf{u}}_n$. This can be done by padding the vector $\mathbf{\Psi}_n(\mathbf{p})$ with zero entries based on the mapping between $\hat{\mathbf{u}}_n$ and $\hat{\mathbf{u}}$. It follows that

$$\hat{u}(\mathbf{p}) = \mathbf{\Psi}(\mathbf{p}) \hat{\mathbf{u}} \tag{12}$$

is equivalent to (9), where $\mathbf{\Psi}(\mathbf{p})$ is a vector with N components and is obtained by inserting $N - n$ zeros into $\mathbf{\Psi}_n(\mathbf{p})$ at the proper places. From (1)–(2) by collocation, we have

$$\mathscr{L}\mathbf{\Psi}(\mathbf{p}_j) \hat{\mathbf{u}} = f(\mathbf{p}_j), \quad j = 1, 2, \ldots, N_I, \tag{13}$$

$$\mathscr{B}\mathbf{\Psi}(\mathbf{p}_j) \hat{\mathbf{u}} = g(\mathbf{p}_j), \quad j = N_I + 1, N_I + 2, \ldots, N. \tag{14}$$

This is an $N \times N$ linear system, and each equation in the system contains n non-zero entries. The system is a sparse system with N unknowns $\{\hat{u}(\mathbf{p}_j)\}_{j=1}^N$. Even though an additional polynomial basis needs to be added when we create local small linear systems in LMAPS using PS, the resulting global sparse system remains the same size as in LMAPS using other positive definite RBFs.

3. RBFs and selection of shape parameters

In the past, MQ was primarily the basis function adopted in LMAPS. Recently, Lamichhane and Chen et al. found [13,11] the analytical particular solutions for Δ and $\Delta - \lambda^2$ with Gaussian and Matérn RBFs. Then LMAPS has been extended to those RBFs as well. In this paper, we will test the numerical performance of the proposed LMAPS with PS and compare to other RBFs listed in Table 1, which includes Gaussian (GA), multiquadric (MQ), and Matérn. Note K_i, $i = 2, 3$ denote the modified Bessel functions of the second kind of order two and three, respectively. We denote the Matérn of order two by Matérn2, and the Matérn of order three by Matérn3, and so on. We denote the polyharmonic splines of order four by PS4, of order five by PS5, and so on. We also denote GA as Gaussian. All of the RBFs listed in Table 1 have shape parameters except PS.

Table 1
List of RBFs used in LMAPS in this paper.

Name of RBFs	Formulation	Shape parameter
Gaussian	$\phi(r) = \exp\left(-cr^2\right),\ \beta > 0$	c
MQ	$\phi(r) = \sqrt{r^2 + c^2},\ c > 0$	c
IMQ	$\phi(r) = 1/\sqrt{r^2 + c^2},\ c > 0$	c
Matérn of order m	$\phi(r) = \begin{cases} (cr)^m K_m(cr), & r > 0 \\ m\Gamma(m), & r = 0 \end{cases}$	c
PS of order m	$\phi(r) = r^{2m}\ln(r)$	None
Thin plate spline	$\phi(r) = r^2 \ln r$	None

Thus, one of the advantages of using LMAPS with PS RBF is that there is no shape parameter involved in the approximation process.

To solve the Poisson equation

$$\Delta u(x, y) = f(x, y) \tag{15}$$

by LMAPS, we need particular solution $\Phi(r)$ of the Laplace operator with respect to the chosen RBF, $\phi(r)$ in the right-hand side:

$$\Delta\Phi(r) = \phi(r). \tag{16}$$

For the modified Helmholtz equation of the form

$$(\Delta - \lambda^2)u(x, y) = f(x, y) \tag{17}$$

there are two main operators involved in LMAPS: the Laplace operator Δ and the modified Helmholtz operator $\Delta - \lambda^2$. We have the choice of using the particular solutions of the Laplace:

$$\Delta\Phi(r) = \phi(r), \tag{18}$$

or the particular solutions of the modified Helmholtz operator:

$$(\Delta - \lambda^2)\Phi(r) = \phi(r). \tag{19}$$

We will compare the numerical performance of the proposed improved LMAPS on both choices. For more details on how the Laplace operator is used in the modified Helmholtz equation, we refer readers to [8]. For most of the other kinds of elliptic equations of second order, for example, the Laplace equation with variable coefficient convection–reaction terms, using the particular solutions of the Laplacian operator in LMAPS will be an easier choice. A list of particular solutions can be found in the Appendix at the end of this paper [13,11,8].

Searching for an optimal shape parameter in RBFs is an outstanding research topic. Despite the efforts in [24–27] and references therein, selection for a suitable shape parameter for RBFs remains a challenge. In this paper, we employ two techniques on the local $n \times n$ linear systems, where n is the number of points in the local influence domains, to find an optimal shape parameter. The first technique is called the leave-one-out-cross-validation (LOOCV) [19,25]. LOOCV is an estimation of the optimal parameter of a model of n samples trained on $n - 1$ samples, which has a high variance. In general, the more collocation points we have, the better shape parameter c can be selected by LOOCV. When n is not too small, we will couple LMAPS with LOOCV to select a good shape parameter of MQ, Gaussian, and Matérn RBFs. However, when n is small, such as $n = 5$, LOOCV will not work well. Then, we will utilize the scaling technique, where the shape parameter is scaled by the maximum nodal distance in the local domains. We refer readers to [28,29] for further details.

4. Numerical results

Throughout this section, the following notations are used: m is the order of PS or the maximum degree of polynomials added to the RBF kernels, n is the number of interpolation points in the domain of influence, N_I is the number of interior points, N_B is the number of boundary points, and $N = N_I + N_B$ is the total number of interpolation points. We compare all of the numerical results with the analytical solutions. We also compare our results of the improved LMAPS with the Localized Kansa's Method (LKM) [30]. We use the maximum relative absolute error ε_1 and the root mean square error ε_2 to measure the numerical accuracy. The definitions of these errors are given as follows:

$$\varepsilon_1 = \max_{k=1}^{N} \left| \frac{\hat{u}_k - u_k}{u_k} \right|$$

$$\varepsilon_2 = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left(\hat{u}_k - u_k\right)},$$

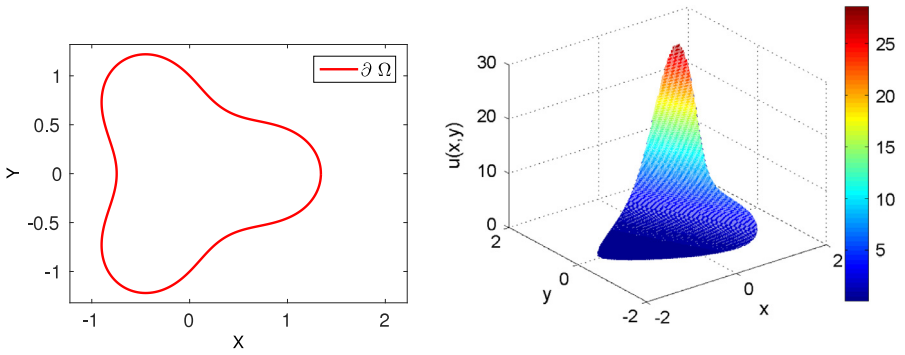

Fig. 1. The profiles of the computational domain and the analytical solution in Example 1.

Table 2
Comparison of PS and MQ for various m using $n = 25$.

m	PS		MQ			
	CPU(s)	ε_2	CPU(s)	ε_2	c	
0	–	–	3.27	2.83×10^{-5}	1.64	OLMAPS
1	5.18	6.92×10^{-3}	4.81	8.61×10^{-5}	1.64	
2	5.50	1.40×10^{-5}	4.07	1.82×10^{-4}	2.19	
3	6.65	1.04×10^{-7}	4.46	5.32×10^{-5}	2.19	
4	6.84	1.91×10^{-9}	5.57	5.29×10^{-6}	3.28	

where $\hat{u}_k = \hat{u}(\mathbf{p}_k)$ is the approximated solution, and $u_k = u(\mathbf{p}_k)$ is the exact solution. To distinguish the performance of the original LMAPS from the improved method, we marked the corresponding results from the original method with OLMAPS or (O), and the results from the improved method with ILMAPS or (I).

Example 1. In this example, we consider the following Poisson problem:

$$\Delta u(x, y) = 5e^{2x+y}, \quad (x, y) \in \Omega, \tag{20}$$

$$u(x, y) = e^{2x+y}, \quad (x, y) \in \partial\Omega \tag{21}$$

where Ω and the analytical solutions are shown in Fig. 1. The boundary $\partial\Omega$ is defined by the following parametric equation:

$$\partial\Omega = \{(x, y) | x = \rho \cos\theta + 1.5,\ y = \rho \sin\theta + 1.5, 0 \le \theta < 2\pi\},$$

where

$$\rho = \left(\cos(3\theta) + \sqrt{2 - \sin^2(3\theta)}\right)^{\frac{1}{3}}.$$

In the numerical implementation in this example, we choose $N_I = 9189$ and $N_B = 300$. In OLMAPS with MQ as shown in [8], there is no need to add a polynomial basis. However, to test the effect of the polynomial basis added to PS in LMAPS, we add a polynomial basis of degree up to m to MQ, and compare it with the numerical results from PS of order m.

Table 2 shows ε_2 when we use PS of order $m = 1$ to $m = 4$, and MQ with polynomial basis of degree up to $m = 4$ and $n = 25$. Note that $m = 0$ in MQ indicates OLMAPS without additional polynomial basis. It produces error $\varepsilon_2 = 2.83 \times 10^{-5}$ which is similar to the results when a polynomial basis of degree up to three is added to MQ. Furthermore, LMAPS with PS of order one performs poorly compared to the others. However, as we further increase the order of PS, the numerical results become far more accurate than MQ. In addition, with the scaling technique, the range of good shape parameters for MQ can be very large. However, we still need to determine what range and what values of the shape parameter produces relatively better results than other values of c, which is time-consuming and inefficient. Unlike MQ, we do not need to search for an optimal shape parameter for PS.

Table 3 shows the results using various RBFs and n in OLMAPS and ILMAPS. Table 4 shows the corresponding condition numbers of global sparse matrices. For RBFs with shape parameter, we use LOOCV to search for a good shape parameter when n is sufficiently large. The initial search interval for LOOCV is $[0, b]$ where $b = 2, 3, 15$ depending on the basis functions and n. We need to be careful how to find appropriate values of b for each case. This is not the focus in this paper, so we simply listed the corresponding optimal shape parameters in the table. When $n = 5$, we use the scaling technique and manually select an acceptable shape parameter after a few experiments. The shape parameters after scaling are listed in Table 3. As we can see, PS of order $m = 4$ outperforms all of the other RBFs. LKM introduced in [30] has similar accuracy as LMAPS when MQ is used (data not shown). We can simply increase the number of points in the local influence domains or the order of PS

Table 3
ε_2 for various n and RBFs.

n	GA(O)	c	MQ(O)	c	Matérn 3(O)	c	PS4(I)
5	2.51×10^{-4}	1.16	2.72×10^{-4}	0.58	3.36×10^{-4}	0.58	–
15	4.04×10^{-5}	0.98	5.30×10^{-5}	0.81	1.82×10^{-5}	2.07	2.40×10^{-5}
25	2.69×10^{-5}	0.98	1.01×10^{-4}	3.75	6.88×10^{-5}	1.99	1.91×10^{-9}
35	3.51×10^{-5}	1.77	2.01×10^{-5}	5.21	3.69×10^{-5}	2.29	7.68×10^{-10}
45	3.64×10^{-5}	1.24	7.53×10^{-5}	1.46	5.82×10^{-4}	1.29	4.76×10^{-10}

Table 4
Condition numbers of the sparse matrices for various n and RBFs.

n	GA(O)	MQ(O)	Matérn 3(O)	PS4(I)
5	2.75×10^{5}	2.34×10^{5}	2.59×10^{5}	–
15	1.43×10^{10}	1.72×10^{10}	5.78×10^{6}	6.78×10^{19}
25	1.65×10^{10}	5.64×10^{9}	1.01×10^{10}	2.21×10^{5}
35	2.11×10^{8}	2.23×10^{9}	5.99×10^{9}	2.82×10^{5}
45	4.49×10^{10}	1.76×10^{10}	8.30×10^{10}	3.54×10^{5}

Table 5
ε_1 with increasing noises on f and g for various RBFs using $n = 25$.

η	GA(O)	c	MQ(O)	c	Matérn 3(O)	c	PS4(I)
1.0	8.81×10^{-2}	4.92	4.46×10^{-2}	0.10	6.08×10^{-2}	2.73	2.51×10^{-2}
0.1	8.04×10^{-3}	4.92	4.91×10^{-2}	0.58	2.85×10^{-3}	2.73	1.50×10^{-3}
0.01	1.35×10^{-3}	4.92	4.59×10^{-2}	0.58	3.83×10^{-4}	2.73	1.00×10^{-4}
0.001	1.44×10^{-3}	3.28	4.09×10^{-3}	1.64	5.55×10^{-5}	2.73	1.04×10^{-5}
0.000	3.62×10^{-4}	4.92	1.02×10^{-3}	0.58	3.31×10^{-6}	2.73	3.49×10^{-9}

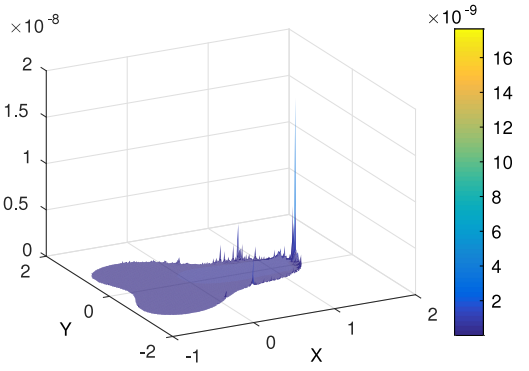

Fig. 2. The absolute error in Example 1 using $m = 4$ and $N = 27\,644$.

to improve the numerical accuracy when PS is utilized. Furthermore, LMAPS using PS with $n = 25$ is very accurate. Further increasing the number of neighboring points in the local influence domains does not cause any ill-conditioning.

From Table 4, it is clear that the condition numbers from PS4 do not increase much when the number of local points increases. However, the RBFs with shape parameter produce much larger condition numbers when n becomes larger. PS is more stable than the other RBFs in terms of accuracy with respect to the number of local points in the influence domains.

One may notice that we did not list any results for LMAPS with PS4 on five neighboring points in the local domains in Table 3. In this case, the small matrices $\mathbf{\Phi}_{nn}$ in (7) are singular. When singular value decomposition (SVD) is used, we can obtain $\varepsilon_2 = 1.2537 \times 10^{-4}$, which is similar in accuracy to the other RBFs. The corresponding condition number is only 1.66×10^{5}. To compare all the RBFs fairly, we used only Gaussian elimination to construct the global sparse matrices in all of the numerical results listed in the tables. Thus, we did not list the better results after SVD is used in ILMAPS with PS4.

Fig. 2 shows the profile of absolute errors when PS4 is used in ILMAPS. The maximum absolute error is in the order of 10^{-8} which is far more accurate than OLMAPS.

Fig. 3 shows the rate of convergence using various RBFs with respect to different total numbers of points, N. OLMAPS with GA, MQ and Matérn3 have similar rates of convergence. ILMAPS with PS4 has order of four convergence approximately, which is more rapid than the others once the number of points in the domain reaches a certain level.

To test the stability using various RBFs, we add noise to the forcing term and the boundary conditions by η%. The maximum relative absolute error, ε_1 is shown in Table 5, where $\eta = 0$ means that noise is not added. The corresponding optimal shape parameters for RBFs with shape parameters are also listed. When η becomes large, all of the RBFs perform the

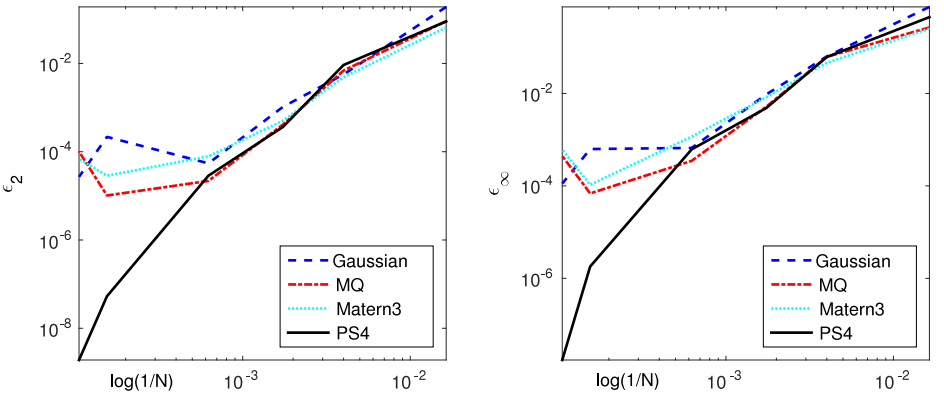

Fig. 3. The rate of convergence of LMAPS for various RBFs and N in Example 1.

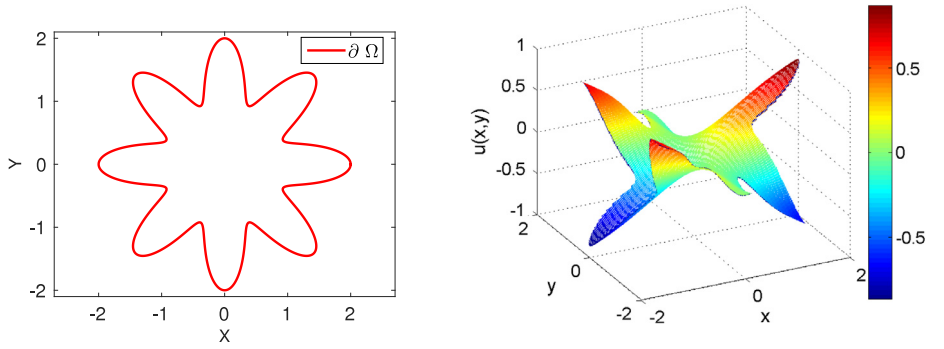

Fig. 4. The profiles of computational domain and the analytical solution in Example 2.

same. When the noise is relatively small, Matérn and PS perform slightly better than the others. Again, we have to search for the best approximation when Matérn is used. In this example, we conclude that PS4 has the best performance in terms of accuracy and stability.

Example 2. In this example, we consider the following modified Helmholtz equation:

$$(\Delta - 100)\,u(x,y) = f(x,y), \quad (x,y) \in \Omega$$
$$u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega,$$

where

$$f(x,y) = \left(-\frac{\pi^2}{36} - 9\frac{\pi^2}{16} - 100\right)\sin\left(\frac{\pi x}{6}\right)\cos\left(\frac{3\pi y}{4}\right), \quad \text{and}$$
$$g(x,y) = \sin\left(\frac{\pi x}{6}\right)\cos\left(\frac{3\pi y}{4}\right).$$

The analytical solution is given by

$$u(x,y) = \sin\left(\frac{\pi x}{6}\right)\cos\left(\frac{3\pi y}{4}\right). \tag{22}$$

The star-shape domain Ω is defined by the following parametric equation:

$$\Omega = \{(x,y) | x = \rho\cos(\theta), y = \rho\sin(\theta), 0 \le \theta < 2\pi\},$$

where

$$\rho = 1 + \cos^2(4\theta).$$

The profiles of the computational domain and the analytical solution are shown in Fig. 4.

In this example we use $N_I = 18\,456$ and $N_B = 1000$ in our numerical implementation. Fig. 5 shows the absolute errors when the particular solution of the Laplace operator with PS4 and $n = 25$ are used. The absolute errors are in order of 10^{-10} which is extremely accurate compared to the other RBFs.

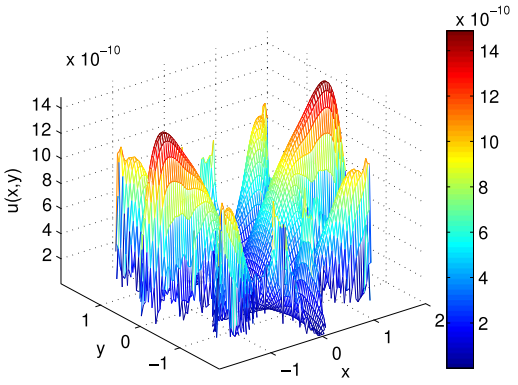

Fig. 5. The absolute error in Example 2 using particular solution for Δ of order $m = 4$.

Table 6
ε_2 of ILMAPS with Δ and $\Delta - \lambda^2$ using PS and $n = 25$.

m	Δ		$\Delta - \lambda^2$	
	CPU(s)	ε_2	CPU(s)	ε_2
1	11.17	8.22×10^{-5}	11.55	1.57×10^{-4}
2	11.92	9.94×10^{-7}	14.87	1.02×10^{-6}
3	13.29	2.00×10^{-7}	19.59	2.25×10^{-7}
4	13.88	5.60×10^{-10}	28.92	3.67×10^{-9}

Table 7
ε_2 of ILMAPS using PS3 for various n.

n	Δ	$\Delta - \lambda^2$
15	2.36×10^{-7}	2.64×10^{-7}
25	2.00×10^{-7}	2.25×10^{-7}
35	9.46×10^{-8}	1.03×10^{-7}
45	7.16×10^{-8}	7.86×10^{-8}
55	5.67×10^{-8}	5.68×10^{-8}
100	2.39×10^{-8}	2.09×10^{-8}

Table 8
ε_2 using various RBFs using Δ and n.

n	GA(O)	c	Matérn 3(O)	c	MQ(O)	c	PS4(I)
5	5.77×10^{-5}	1.86	1.15×10^{-4}	4.05	8.12×10^{-5}	1.86	–
15	3.08×10^{-7}	4.05	1.01×10^{-6}	1.35	2.97×10^{-7}	0.45	2.41×10^{-4}
25	2.02×10^{-6}	5.72	1.04×10^{-6}	1.91	9.04×10^{-7}	0.31	5.60×10^{-9}
35	3.51×10^{-7}	7.24	1.51×10^{-6}	7.24	3.74×10^{-7}	0.31	2.30×10^{-10}
45	7.38×10^{-7}	8.09	1.38×10^{-6}	8.09	1.63×10^{-6}	0.17	1.50×10^{-10}

Table 6 shows ε_2 using the Laplace operator Δ and the modified Helmholtz operators $\Delta - \lambda^2$ with PS of different orders and $n = 25$. The computational time using either operator is relatively close. The numerical accuracy is similar when $m < 4$. Our experiments in this table show that the particular solution w.r.t. the Laplace operator in the modified Helmholtz equation produces relatively stable numerical approximations.

Table 7 shows ε_2 using particular solutions of PS3 for different operators with various n. We notice that there is a little difference in terms of accuracy. Since the particular solutions for the modified Helmholtz operator using MQ, GA, and Matérn are not available, particular solutions of the Laplace operator are used in our following numerical tests.

Table 8 shows ε_2 using different RBFs, including GA, Matérn3, MQ, and PS4 with various n. We tried our best to find 'the best' shape parameters for RBFs other than PS4 and they are listed in Table 8. Notice that GA and Matérn3 are relatively stable in terms of the best shape parameters. We used the scaling technique to find the best shape parameters, where GA and Matérn3 do not require much of the tests before we found the 'best' shape parameter. On the other hand, MQ is more sensitive to the shape parameters. We do not need to use any optimization technique for PS. We can simply increase the order of the basis, or the number of points in the local domains to increase the accuracy. With $n = 15$, PS4 is less accurate compared to any other RBF. However, PS3 could easily outperform other RBFs with $n = 15$ (data not shown). When n becomes larger, PS4 outperforms all of the other basis functions by more than three order of magnitudes.

Table 9
Condition numbers and computational time using various RBFs with Δ for various n.

n	Cond. #				CPU(s)			
	GA(O)	Matérn 3(O)	MQ(O)	PS4(I)	GA(O)	Matérn 3(O)	MQ(O)	PS4(I)
5	1.54×10^4	1.41×10^4	6.60×10^3	–	5	9	3	–
15	1.28×10^4	3.00×10^5	4.13×10^4	2.05×10^{16}	8	18	5	9
25	1.76×10^4	1.95×10^5	8.40×10^8	2.08×10^4	19	30	10	14
35	2.46×10^5	1.99×10^4	1.03×10^9	2.48×10^4	24	49	10	21
45	3.66×10^7	1.91×10^4	4.63×10^4	2.37×10^4	39	75	12	29

Table 10
ε_2 with increasing noises on f and g for various RBFs using $n = 25$.

η	GA(O)	c	MQ(O)	c	Matérn 3(O)	c	PS4(I)
1.0	1.81×10^{-3}	7.56	1.71×10^{-2}	13.71	4.64×10^{-3}	5.65	1.71×10^{-3}
0.1	1.66×10^{-4}	7.56	1.58×10^{-2}	0.57	8.04×10^{-4}	5.65	1.69×10^{-4}
0.01	1.00×10^{-5}	7.56	1.83×10^{-3}	0.57	4.25×10^{-4}	5.65	1.72×10^{-5}
0.001	8.84×10^{-6}	7.56	2.15×10^{-4}	0.57	3.86×10^{-4}	2.93	1.68×10^{-6}
0	3.42×10^{-6}	3.63	3.21×10^{-6}	0.57	4.21×10^{-5}	2.93	5.41×10^{-10}

As we increase the number of neighboring points in the local influence domains, the numerical results using GA, Matérn3 and MQ remain at similar accuracy. However, PS4 has significantly better results than those basis functions containing a shape parameter. Even with $n = 100$ neighboring points in the local domains, PS4 has no stability issues (data not shown). Thus, if we require to use a larger n, it is suggested to use a higher-order of PS such as $m = 4$ with $n > 15$. We also tested LKM on the modified Helmholtz problem. The numerical accuracy of LKM is similar to LMAPS with MQ using the Laplace operator (data not shown).

The condition numbers of the global sparse matrices and computational time of the tests in Table 8 are listed in Table 9. The condition numbers from PS4 are very stable once the number of local points reaches 25. However, when $n = 15$ the condition number of the large sparse matrix by PS4 is quite large. As a result, the accuracy of PS4 is not as good as other RBFs for $n = 15$. In fact, for $n < 15$, the condition numbers are high and the proposed method is not working well. However, when $n > 25$, PS4 is much more accurate than the other RBFs, and the condition number remains low when we further increase the number of local points. In addition, further increasing n will not cost any loss of accuracy. The computational time for PS4 is not much more than GA, but relatively large compared to MQ. This is due to the augmented polynomials we added to the basis. We used the symbolic math tools in MATLAB to define the augmented polynomials and compute their derivatives in our simulation. It is an easier way than defining polynomial basis term by term but it is also time-consuming. The reported computational time in the manuscript is from the code with symbolic computation. It is possible to elimination of the symbolic computations for reducing computational time.

Table 10 shows the accuracy using various RBFs when η% noise is added to the forcing term and the boundary condition. When noise is 1%, there is not much difference between all of the RBFs tested. When η becomes smaller, PS and GA have similar accuracy while MQ and Matérn3 are slightly worse. Our first example shows that PS and Matérn outperform the other RBFs containing shape parameters. In both cases, PS is always one of the best choices, not to mention that there is no need to search for optimal shape parameters. Thus, LMAPS is more stable when PS is utilized.

Fig. 6 shows the rate of convergence using various RBFs with respect to different numbers of interpolation points, N. Similar to our previous example, LMAPS with GA, MQ and Matérn4 have similar rates of convergence, approximately linear convergence. However, LMAPS with PS4 has an order of four convergent rate, which is much superior to other RBFs.

Example 3. In this example, we consider the following variable coefficient elliptic equation with advection–reaction terms

$$\Delta u + x^2 y u + y^3 \sin(x) \frac{\partial u}{\partial x} - y^2 \cos(x) \frac{\partial u}{\partial y} = f(x, y), \quad (x, y) \in \Omega \tag{23}$$

$$u(x, y) = y \cos(x) + x \cos(y), \quad (x, y) \in \partial\Omega \tag{24}$$

where $f(x, y)$ is given according to the analytical solution, which is given as

$$u(x, y) = y \cos(x) + x \cos(y). \tag{25}$$

The domain is defined by the following parametric equation

$$\partial\Omega = \{(x, y), x = \rho \cos(\theta), y = \rho \sin(\theta), 0 \le \theta \le 2\pi\}, \tag{26}$$

where

$$\rho = e^{\sin(\theta)} \sin^2(2\theta) + e^{\cos(\theta)} \cos^2(2\theta). \tag{27}$$

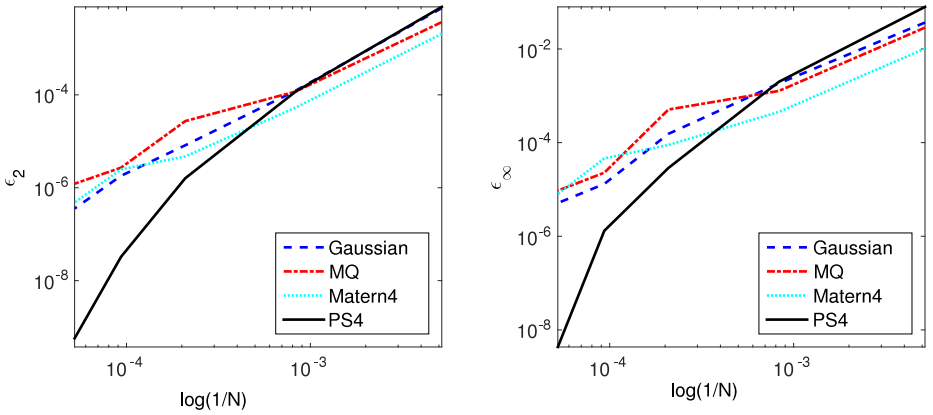

Fig. 6. The rate of convergence of LMAPS with various RBFs and N in Example 2.

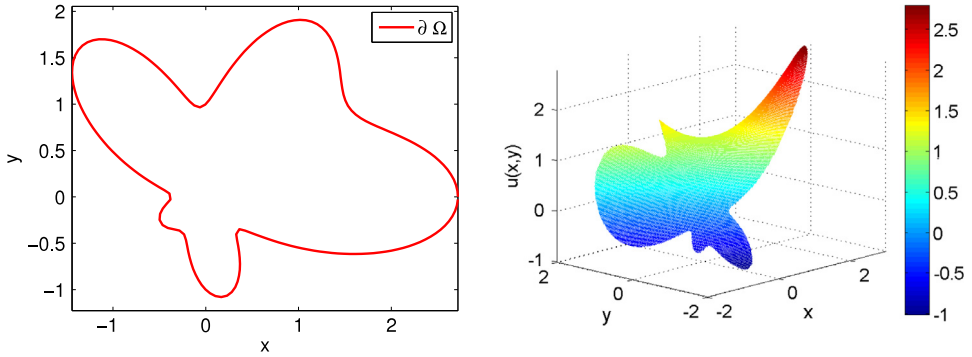

Fig. 7. The profiles of the computational domain and the analytical solution in Example 3.

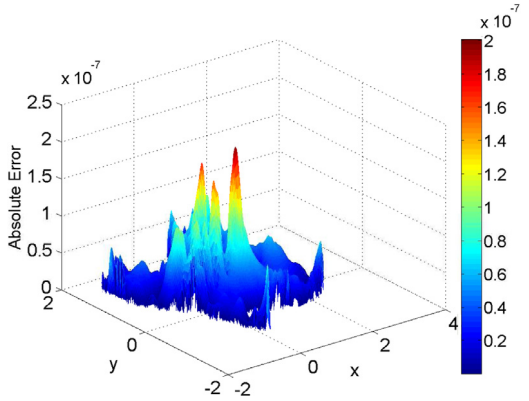

Fig. 8. The absolute error using LMAPS with PS5 in Example 3.

The profiles of the amoeba-like domain and the analytical solution are shown in Fig. 7. Fig. 8 shows the absolute errors using PS5 and $n = 55$. The accuracy reaches an order of 10^{-7}.

However, OLMAPS with GA, MQ or Matérn cannot approximate the solution correctly. On the other hand, ILMAPS with PS can produce acceptable numerical solutions when the order of PS is high enough. We also add the polynomial basis of degree up to five to the other RBFs, which makes ILMAPS produce acceptable accuracy with GA, MQ or Matérn. Therefore, ILMAPS, with all the RBFs considered in this paper, can be applied to the elliptic PDEs with variable coefficients. Thus, we conclude the additional polynomial basis made OLMAPS possible to solve complicated elliptic equations with improved accuracy in this example.

Fig. 9 shows the rate of convergence of ILMAPS for various RBFs. ILMAPS is relatively stable with MQ and PS5. Furthermore, ILMAPS with PS5 outperforms all of these RBFs.

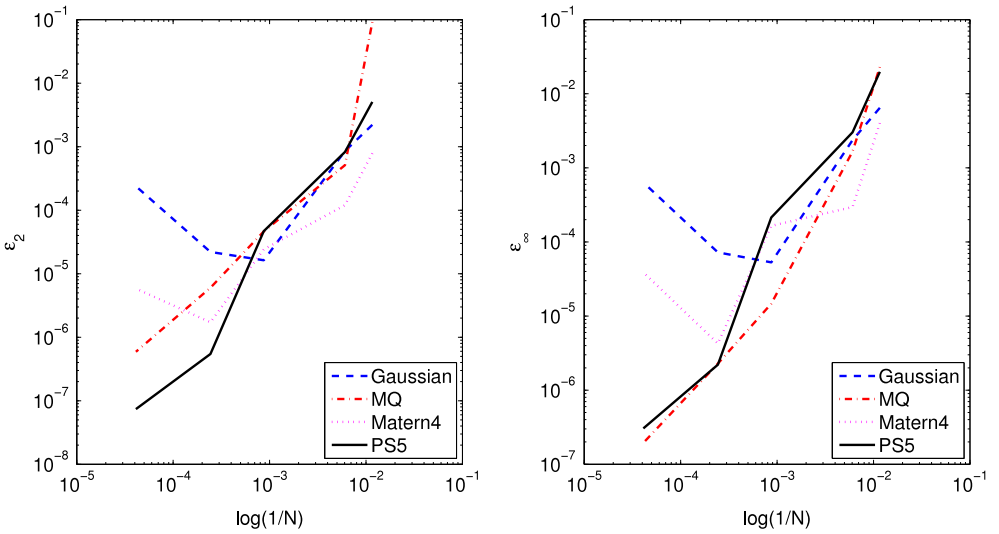

Fig. 9. The rate of convergence of LMAPS for various RBFs in Example 3.

Table 11
ε_2 of ILMAPS with Δ using various orders of PS and $N_I = 23\,851$, $N_B = 500$ in Example 3.

m	CPU(s)	ε_2	Condition number	n
1	10	3.21×10^{-2}	1.09×10^{6}	5
2	14	5.04×10^{-2}	8.62×10^{12}	10
3	47	4.84×10^{-3}	3.45×10^{12}	35
4	52	4.03×10^{-5}	3.77×10^{12}	35
5	52	7.40×10^{-8}	4.54×10^{11}	35

Table 12
ε_2 of ILMAPS using various RBFs, n and $N_I = 23\,851$, $N_B = 500$ in Example 3.

n	GA(I)	c	Matérn 4(I)	c	MQ(I)	c	PS5(I)
35	2.19×10^{-4}	5.00	5.84×10^{-6}	2000	5.91×10^{-7}	105.22	7.40×10^{-8}
45	1.81×10^{-4}	5.00	4.11×10^{-6}	2000	9.08×10^{-8}	119.98	6.75×10^{-8}
55	3.55×10^{-4}	5.00	4.06×10^{-6}	2000	1.58×10^{-7}	137.20	4.18×10^{-8}
65	1.32×10^{-4}	0.65	2.20×10^{-6}	2000	1.94×10^{-7}	223.22	1.58×10^{-7}

Table 13
Condition numbers and computational time of the sparse matrices using various RBFs, n, and $N_I = 23\,851$, $N_B = 500$ in Example 3.

n	Cond. #				Time (s)			
	GA(I)	Matérn4(I)	MQ(I)	PS5(I)	GA(I)	Matérn 4(I)	MQ(I)	PS5(I)
35	3.03×10^{10}	8.96×10^{9}	3.88×10^{11}	4.54×10^{11}	60	158	36	52
45	6.04×10^{11}	8.63×10^{9}	6.18×10^{10}	7.28×10^{12}	84	233	45	79
55	2.63×10^{11}	5.36×10^{10}	3.16×10^{10}	4.56×10^{12}	108	297	52	81
65	2.61×10^{10}	2.83×10^{9}	1.09×10^{10}	8.50×10^{12}	107	400	71	94

Table 11 shows ε_2, the corresponding computational time, condition numbers of the global sparse matrices using PS of different orders, and different numbers of points in the local domains. The numerical accuracy becomes acceptable when the order of PS $\geq$ 3. Compared to the previous two examples, we observe that we need to increase the number of local points in this example. This is reasonable as we need more points in the local domains to support the approximation of the partial derivatives. As shown in Table 11, when $n \geq 35$, the numerical results become very accurate. In our test, it would not improve the accuracy if we continue to increase the number of the local points n. On the other hand, when the total number of interpolation points is chosen as $N = 4309$, further increasing n to 65 would decrease the accuracy. In this case, the condition number of the global sparse matrix becomes fairly large as we will see in Table 14. Thus, it is suggested to use a small n in the local influence domains, such as $n = 35$ in PS5.

Tables 12–15 show ε_2, the corresponding computational time, condition numbers using GA, MQ and Matérn of order four, and n points in the local domains. Tables 12–13 are the results using $N = 24\,351$. Tables 14–15 are the results using

Table 14
ε_2 of ILMAPS using various RBFs, n, and $N_I = 3809$, $N_B = 500$ in Example 3.

n	GA(I)	c	Matérn4(I)	c	MQ(I)	c	PS5(I)
35	1.10×10^{-4}	5.00	4.86×10^{-7}	500	8.98×10^{-7}	118.38	7.07×10^{-7}
45	2.34×10^{-5}	5.00	4.71×10^{-7}	250	9.05×10^{-7}	44.99	6.58×10^{-7}
55	4.82×10^{-5}	5.00	4.99×10^{-7}	200	2.16×10^{-6}	51.45	4.55×10^{-6}
65	5.28×10^{-5}	5.00	4.74×10^{-7}	100	1.82×10^{-6}	55.80	4.19×10^{-4}

Table 15
Condition numbers and computational time of the sparse matrices using various RBFs, Δ, n, and $N_I = 3809$, $N_B = 500$ in Example 3.

n	Cond. #				CPU (s)			
	GA(I)	Matérn4(I)	MQ(I)	PS5(I)	GA(I)	Matérn4(I)	MQ(I)	PS5(I)
35	1.05×10^{8}	2.03×10^{6}	7.50×10^{8}	1.72×10^{10}	8	27	5	12
45	1.52×10^{9}	2.41×10^{6}	9.28×10^{8}	3.79×10^{11}	16	40	6	13
55	6.91×10^{8}	5.72×10^{6}	5.93×10^{8}	1.35×10^{12}	21	51	7	13
65	3.92×10^{9}	3.33×10^{6}	4.43×10^{8}	2.28×10^{14}	29	74	8	15

$N = 4309$. It is obvious that LMAPS with PS5 outperforms all of the other basis functions, not to mention the tedious and expensive numerical process of seeking the optimal shape parameters for the other RBFs.

OLMAPS with Gaussian and Matérn performs poorly without polynomial basis. OLMAPS using MQ without augmented polynomials has low accuracy for solving elliptic PDE with variable coefficients. With augmented polynomial basis, especially polynomials of order five, MQ performs as well as PS5 in ILMAPS. The computational costs among four RBFs are not as similar to each other as it was in our previous examples. However, the computational time using PS5 is only slightly longer than MQ. We tested the sensitivity of ILMAPS to the noise on the forcing term and on the boundary. Unfortunately we are not able to produce very good results ($\varepsilon_2 \approx 10^{-1}$ with $\eta = 0.001$) from any of the RBFs considered. Our understanding is that this is a complicated variable coefficient PDE on a complicated computational domain. It is very challenging to even approximate the solution accurately in general.

5. Conclusion

In this paper, the polyharmonic splines (PS) have been first utilized in the context of LMAPS. A low degree of polynomial basis in added to insure the invertibility of the local systems in LMAPS. Unlike Gaussian, MQ or Matérn RBFs, there is no need to search for optimal shape parameter in PS. The numerical accuracy of ILMAPS using PS can be improved by simply increasing the number of points in the local influence domains, or increasing the order of PS RBFs.

We examined the numerical results of the proposed approach through three typical elliptic PDEs: a Poisson equation, a modified Helmholtz equation, and a general variable coefficient second-order elliptic PDE on irregular domains. Numerical tests show that ILMAPS with PS is far superior than using any other RBFs. This is a surprise and is unexpected.

The improved LMAPS with PS on the Poisson equation has much better accuracy than other high performance RBFs at the order five magnitude which is quite significant. The numerical accuracy of ILMAPS with PS is better than LKM [30].

For the modified Helmholtz equations, we have the choices in LMAPS to use particular solutions of the Laplace operator or the modified Helmholtz operator. We compared the numerical performance of the improved LMAPS on both operators. The experimental results suggest that the improved LMAPS performs well with both operators, though the Laplace operator performs slightly better than the modified Helmholtz operator.

For the general variable coefficient second-order elliptic equation, OLMAPS with MQ works on such problems but with low accuracy. With any other basis function, the original LMAPS failed to produce reasonable results of such problems on the irregular domains. On the other hand, ILMAPS with all of the RBFs work well on such problems. In some cases, MQ can even compete with PS on accuracy and efficiency if we can find the optimal shape parameter. Therefore, ILMAPS with PS remains to be the most accurate, efficient, and stable RBF comparing to other RBFs.

The proposed improved approach offers the prospect of an efficient algorithm for solving more challenging problems in science and engineering. Advantages using the improved LMAPS includes: 1. There is no need to search for good shape parameters for acceptable numerical results when utilizing PS. This makes the method efficient compared to other RBFs with shape parameters; 2. We can improve the accuracy by simply increasing the order of polyharmonic splines, or the number of interpolation points; With an order up to 20° polynomials, we can obtain amazingly high accuracy without any ill-conditioning issues; 3. With additional polynomial basis, LMAPS becomes more stable. However, we also face some challenges: 1. The symbolic math tool in MATLAB is time-consuming; 2. Small influence domain is not working well using polyharmonic splines. We can improve the computational efficiency further by eliminating the symbolic computation. Therefore, computational time for adding more influence points in the local influence domains will remain in a reasonable range. Our ongoing research will use the time-stepping method or the higher-order discretization scheme in time space to solve diffusion–reaction–convection equations.

Table 16
List of particular solutions $\Phi(r)$ for Δ using different RBFs $\phi(r)$, where $\Delta\Phi(r) = \phi(r)$.

RBF $\phi(r)$	Kernel $\Phi(r)$
Gaussian	$\begin{cases} \frac{1}{4c} Ei(cr^2) + \frac{1}{2c}\log(r), & r \neq 0 \\ \frac{1}{4c}(-0.5772 + \log(c)), & r = 0 \end{cases}$
MQ	$\frac{1}{9}\left(4c^2 + r^2\right)\sqrt{r^2 + c^2} - \frac{c^3}{3}\ln\left(c + \sqrt{r^2 + c^2}\right)$
Matérn of order m	$\begin{cases} \frac{m!}{c^2}\sum_{j=0}^{m}\left(\frac{2^j}{(m-j)!}(cr)^{m-j}K_{m-j}(cr) + 2^m\ln(cr),\right), & r \neq 0 \\ \frac{m!2^m}{c^2}\left(\ln(2) - \gamma + \sum_{j=0}^{m-1}\frac{1}{(m-j)!}\right), & r = 0 \end{cases}$
PS of order m	$\frac{r^{2m+2}\ln r}{4(m+1)^2} - \frac{r^{2m+2}}{4(m+1)^3}$

Table 17
List of derivatives of particular solutions for Δ using different RBFs, $\frac{d\Phi(r)}{rdr}$.

Type of RBF	Kernel $\frac{\partial\Phi(r)}{r\partial r}$
Gaussian	$\begin{cases} \frac{-e^{-cr^2}+1}{2r^2c}, & r \neq 0 \\ \frac{1}{2}, & r = 0 \end{cases}$
MQ	$\frac{1}{3}\frac{c\sqrt{r^2+c^2}+r^2+2c^2}{c+\sqrt{r^2+c^2}}$
Matérn of order m	$\begin{cases} -(cr)^{m-1}K_{m+1}(cr) + \frac{2^m m!}{(cr)^2}, & r \neq 0 \\ 2^{m-2}(m-1)!, & r = 0 \end{cases}$
Kansa	$\frac{1}{\sqrt{r^2+c^2}}$

Table 18
List of particular solutions for $\Delta - \lambda^2$ using PS RBFs [31].

RBF $\varphi(r)$	Kernel $\Phi(r)$
$r^2\ln r$	$\begin{cases} -\frac{4}{\lambda^4}\left(K_0(\lambda r) + \ln(r)\right) - \frac{r^2\ln r}{\lambda^2} - \frac{4}{\lambda^4}, & r > 0 \\ \frac{4}{\lambda^4}\left(\gamma + \ln(\frac{\lambda}{2})\right) + \frac{4}{\lambda^4}, & r = 0 \end{cases}$
$r^4\ln r$	$\begin{cases} -\frac{64}{\lambda^6}\left(K_0(\lambda r) + \ln(r)\right) - \frac{r^2\ln r}{\lambda^2}\left(\frac{16}{\lambda^2} + r^2\right) - \frac{8r^2}{\lambda^4} - \frac{96}{\lambda^6}, & r > 0 \\ -\frac{64}{\lambda^6}\left(\gamma + \ln\left(\frac{\lambda}{2}\right)\right) - \frac{96}{\lambda^6}, & r = 0 \end{cases}$
$r^6\ln r$	$\begin{cases} -\frac{2304}{\lambda^8}\left(K_0(\lambda r) + \ln(r)\right) - \frac{r^2\ln r}{\lambda^2}\left(\frac{576}{\lambda^4} + \frac{36r^2}{\lambda^2} + r^4\right) - \frac{12r^2}{\lambda^4}\left(\frac{40}{\lambda^2} + r^2\right) - \frac{4224}{\lambda^8}, & r > 0 \\ -\frac{2304}{\lambda^8}\left(\gamma + \ln\left(\frac{\lambda}{2}\right)\right) - \frac{4224}{\lambda^8}, & r = 0 \end{cases}$
$r^6\ln r$	$\begin{cases} -\frac{2304}{\lambda^8}\left(K_0(\lambda r) + \ln(r)\right) - \frac{r^2\ln r}{\lambda^2}\left(\frac{576}{\lambda^4} + \frac{36r^2}{\lambda^2} + r^4\right) - \frac{12r^2}{\lambda^4}\left(\frac{40}{\lambda^2} + r^2\right) - \frac{4224}{\lambda^8}, & r > 0 \\ -\frac{2304}{\lambda^8}\left(\gamma + \ln\left(\frac{\lambda}{2}\right)\right) - \frac{4224}{\lambda^8}, & r = 0 \end{cases}$
$r^8\ln r$	$\begin{cases} -\frac{147456}{\lambda^{10}}\left(K_0(\lambda r) + \ln(r)\right) - \frac{r^2\ln r}{\lambda^2}\left(\frac{36864}{\lambda^6} + \frac{2304r^2}{\lambda^4} + \frac{64r^2}{\lambda^2} + r^6\right) - \frac{r^2}{\lambda^4}\left(\frac{39936}{\lambda^4} + \frac{1344r^2}{\lambda^2} + 16r^4\right) - \frac{307200}{\lambda^{10}}, & r > 0 \\ \frac{147456}{\lambda^{10}}\left(\gamma + \ln\left(\frac{\lambda}{2}\right)\right) - \frac{307200}{\lambda^{10}}, & r = 0 \end{cases}$
$r^{10}\ln r$	$\begin{cases} -\frac{14745600}{\lambda^{12}}\left(K_0(\lambda r) + \ln(r)\right) - \frac{r^2\ln r}{\lambda^2}\left(\frac{3686400}{\lambda^8} + \frac{230400r^2}{\lambda^6} + \frac{6400r^4}{\lambda^4} + \frac{100r^6}{\lambda^2} + r^8\right) \\ -\frac{r^2}{\lambda^4}\left(\frac{4730880}{\lambda^6} + \frac{180480r^2}{\lambda^4} + \frac{2880r^4}{\lambda^2} + 20r^6\right) - \frac{33669120}{\lambda^{12}}, & r > 0 \\ \frac{14745600}{\lambda^{12}}\left(\gamma + \ln\left(\frac{\lambda}{2}\right)\right) - \frac{33669120}{\lambda^{12}}, & r = 0 \end{cases}$

Appendix

We list all of the particular solutions used in our numerical comparison for different differential operators and different RBFs.

Appendix A. Particular solutions for Δ using different RBFs

Table 16 shows a list of particular solutions for Δ using different RBFs, including GA, MQ Matérn order of m and PS of order m. Table 17 shows the corresponding derivatives $\frac{d\Phi(r)}{rdr}$. Note that

$$\frac{\partial \Phi(r)}{\partial x} = \frac{xd\Phi(r)}{rdr}, \qquad \frac{\partial \Phi(r)}{\partial y} = \frac{xd\Phi(r)}{rdr}.$$

For the boundary conditions other than Dirichlet or PDEs with lower order derivatives $\frac{\partial \Phi(r)}{\partial x}$ or $\frac{\partial \Phi(r)}{\partial y}$, we would need the derivatives listed in Table 17.

Appendix B. Particular solutions for $\Delta - \lambda^2$ using PS

Table 18 shows the particular solutions for the modified Helmholtz operator $\Delta - \lambda^2$ when PS is used. These particular solutions are specifically used in Example 2 to test the effect of choices of the differential operators in the Modified Helmholtz equations.

References

[1] C.S. Chen, C.M. Fan, P.H. Wen, The method of particular solutions for solving certain partial differential equations, Numer. Methods Partial Differential Equations 28 (2012) 506–522.
[2] P.H. Wen, C.S. Chen, The method of particular solutions for solving scalar wave equations, Int. J. Numer. Methods Biomed. Eng. 26 (2010) 1878–1889.
[3] C.S. Chen, Y.C. Hon, R.A. Schaback, Scientific computing with radial basis functions. Tech. Rep., Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS 39406, USA, 2005, preprint.
[4] N. Mai-Duy, T. Tran-Cong, Indirect RBFN method with thin plate splines for numerical solution of differential equations, CMES Comput. Model. Eng. Sci. 4 (2003) 85–102.
[5] B. Sarler, T. Tran-Cong, C.S. Chen, Meshfree direct and indirect local radial basis function collocation formulations for transport phenomena, in: A. Kassab, C.A. Brebbia, E. Divo, D. Poljak (Eds.), Boundary Elements XXVII, 2015.
[6] N. Mai-Duy, T. Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, Internat. J. Numer. Methods Engrg. 23 (2001) 1807–1829.
[7] Yao Guangming, Local radial basis function methods for solving partial differential equations (Ph.D. thesis), University of Southern Mississippi, 2010.
[8] Guangming Yao, Joseph Kolibal, C.S. Chen, A localized approach for the method of approximate particular solutions, Comput. Math. Appl. 61 (2011) 2376–2387.
[9] C.A. Bustamante, H. Power, Y.H. Sua, W.F. Florez, A global meshless collocation particular solution method (integrated Radial Basis Function) for two-dimensional Stokes flow problems, Appl. Math. Model. 37 (6) (2013) 4538–4547.
[10] Chia-Ming Fan, Chi-Hung Yang, Wei-Shiang Lai, Numerical solutions of two-dimensional flow fields by using the localized method of approximate particular solutions, Eng. Anal. Bound. Elem. 57 (2015) 47–57.
[11] A.R. Lamichhane, C.S. Chen, The closed-form particular solutions for Laplace and biharmonic operators using a Gaussian function, Appl. Math. Lett. 46 (2015) 50–56.
[12] Xueying Zhang, Haiyan Tian, Wen Chen, Local method of approximate particular solutions for two-dimensional unsteady Burgers equations, Comput. Math. Appl. 66 (12) (2014) 2425–2432.
[13] A.R. Lamichhane, C.S. Chen, Particular solution of Laplace and Bi-harmonic operators using Matérn radial basis function, Appl. Math. Lett. 46 (2015) 50–56.
[14] E.J. Kansa, Multiquadric—a scattered data approximation scheme with applications to computational fluid-dynamics. 2. Solutions to parabolic, hyperbolic and elliptic partial-differential equations, Comput. Math. Appl. 19 (1990) 147–161.
[15] A.I. Tolstykh, D.A. Shirobokov, On using radial basis functions in a "finite difference" mode with applications to elasticity problems, Comput. Mech. 33 (2003) 68–79.
[16] H.Y. Tian, S. Reutskiy, C.S. Chen, A basis function for approximation and the solution of partial differential equations, Numer. Partial Differential Equations 24 (2008) 1018–1036.
[17] M.D. Buhmann, Radial Basis Functions, Theory and Implementations, Cambridge University Press, 2003.
[18] G.E. Fasshauer, Solving partial differential equations by collocation with radial basis functions, in: A.L. Mehaute, C. Rabut, L.L. Schumaker (Eds.), Surface Fitting and Multiresolution Methods, 1997, pp. 131–138.
[19] G.E. Fasshauer, J.G. Zhang, On choosing optimal shape parameters for RBF approximation, Numer. Algorithms 45 (2007) 345–368.
[20] R. Franke, Scattered data interpolation: tests of some methods, Math. Comp. 48 (1982) 181–200.
[21] M.D. Buhmann, Multivariate cardinal interpolation with radial basis functions, Constr. Approx. 6 (1990) 225–255.
[22] J. Duchon, Fonctions-spline du type plaque mince en dimension, Technical Report 231, University of Grenoble, 1975.
[23] J. Duchon, Fonctions-spline energy invariate par rotation, Technical Report 27, University of Grenoble, 1976.
[24] C.-S. Huang, C.F. Lee, A.H.-D. Cheng, Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method, Eng. Anal. Bound. Elem. 31 (7) (2007) 614–623.
[25] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comput. Math. 11 (2–3) (1999) 193–210.
[26] C.H. Tsai, J. Kolibal, Ming Li, The golden section search algorithm for finding a good shape parameter for meshless collocation methods, Eng. Anal. Bound. Elem. 34 (2010) 738–746.
[27] J. Wertz, E.J. Kansa, L. Ling, The role of the multiquadric shape parameters in solving elliptic partial differential equations, Comput. Math. Appl. 51 (8) (2006) 1335–1348.
[28] B. Sarler, R. Vertnik, Meshfree explicit local radial basis function collocation method for diffusion problems, Comput. Math. Appl. (2006) 1269–1282.
[29] Robert Vertnik, Local collocation method for phase-change problems (Master's thesis), University of Nova Gorica, 2007.
[30] C.K. Lee, X. Liu, S.C. Fan, Local multiquadric approximation for solving boundary value problems, Comput. Mech. 30 (2003) 396–409.
[31] A.S. Muleshkov, M.A. Golberg, C.S. Chen, Particular solutions of Helmholtz-type operators using higher order polyhrmonic splines, Comput. Mech. 23 (1999) 411–419.