Computers and Mathematics with Applications 61 (2011) 2376–2387

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

A localized approach for the method of approximate particular solutions

Guangming Yao, Joseph Kolibal, C.S. Chen [*]

Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS 39406, USA

ARTICLE INFO

Article history:
Received 27 June 2010
Received in revised form 6 February 2011
Accepted 7 February 2011

Keywords:
LMAPS
GMAPS
Kd-tree
Sparse system
Particular solutions

ABSTRACT

The method of approximate particular solutions (MAPS) has been recently developed to solve various types of partial differential equations. In the MAPS, radial basis functions play an important role in approximating the forcing term. Coupled with the concept of particular solutions and radial basis functions, a simple and effective numerical method for solving a large class of partial differential equations can be achieved. One of the difficulties of globally applying MAPS is that this method results in a large dense matrix which in turn severely restricts the number of interpolation points, thereby affecting our ability to solve large-scale science and engineering problems.

In this paper we develop a localized scheme for the method of approximate particular solutions (LMAPS). The new localized approach allows the use of a small neighborhood of points to find the approximate solution of the given partial differential equation. In this paper, this local numerical scheme is used for solving large-scale problems, up to one million interpolation points. Three numerical examples in two-dimensions are used to validate the proposed numerical scheme.

1. Introduction

The method of approximate particular solutions (MAPS) has been recently developed to solve various types of partial differential equations [1]. Radial basis functions have been used as a major tool to obtain a closed form approximate particular solution which can be used as a basis function for the approximation of the solution to a given partial differential equation. Previously, after we obtain the particular solution, we convert the given differential equation to a homogeneous equation which can be solved by standard boundary methods, e.g., the boundary knot method, the boundary particle method, and others [2–5]. This is a two-stage approach which is well-known in the literature.

Recently, a new approach has been proposed to use the approximate particular solution to satisfy the differential equation and boundary conditions simultaneously as a one-stage approach [6,7]. These numerical schemes constitute very simple and very effective integrated radial basis function schemes. Along these lines, it is of interest to note that Mai-Duy and Tran-Cong [8] had also developed a so-called indirect RBFN method where the integration is carried out along the axes of the coordinate system, while the integration in the MAPS is carried out in the radial direction due to the radial symmetry of the selected differential operator. We note that the indirect RBFN method is also very effective and powerful for solving numerical PDEs.

Due to the global nature of the scheme, however, the resulting matrix is dense and ill-conditioned, substantially limiting the use of the technique to solving problems with only a limited number of interpolation points. Methods such as the finite element method and the finite difference method are capable of solving large-scale problems due to their localized formulations. Similar to these traditional numerical methods, meshless methods with localized formulations [9–13] are very effective. The extension of the recent developed MAPS to solving large-scale problems is the main purpose of this paper.

[*] Corresponding author.
E-mail addresses: cschen.math@gmail.com, cs.chen@usm.edu (C.S. Chen).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.02.007

In this paper we adopt the idea of LMQ [10] in regard to using a localized formulation and apply it to the MAPS. In this approach the solution of the given partial differential equation is expressed as the linear combination of some proper basis functions, however, instead of finding the weighting functions required in the global approach, we seek the numerical solutions at the interpolation points in the local approach. Furthermore, in the local scheme, due to the collocation approach, no numerical integration is required.

Typically, the collocation approach for the global method is constructed by taking into consideration large number of collocation points in the entire domain. It is generally unstable due to the dense matrix formulation, sensitive to the choice of the free parameters in RBFs, and difficult to solve for large-scale problems if not impossible. There are several methods to circumvent this issue such as domain decomposition [14], the greedy algorithm [15], multi-grid approach and compactly supported RBFs [16], extended precision arithmetic [17], the improved truncated singular valued decomposition [18], the localized formulations [10,11,13], iterative methods [19], fast multipole expansion techniques [20], etc. These approaches all represent a substantial complication of the original simple method, on the one hand, with a very limited advantages on the other hand. The localized formulations can alleviate the ill-posed problem of coefficient matrix. The iterative methods can promote the efficiency in determining the coefficients. The fast multipole expansion techniques can accelerate the calculation of influence matrix. Since only the nearby collocation points are needed in the formulation using the local method, the usual ill-conditioning associated with large, dense matrix systems can be alleviated. Another important advantage of considering a local approach using radial basis functions is that the shape parameter associated with the MQ RBF is only slowly varying. In general, MQ is regarded as one of the best RBFs in terms of accuracy if we can identify a suitable shape parameter. For the global approach, this is still a challenging issue [21–23]. One further advantage of the proposed approach is that the computational efficiency achievable by the localized approach does not compromise the accuracy of the method. Instead of solving a dense matrix in the global approach, the local approach results in a sparse matrix that can be solved efficiently.

The structure of the paper is as follows. In Section 2, we briefly review the method of approximate particular solutions (MAPS). Various types of radial basis functions have been introduced for the formulation of the MAPS. In Section 3, we extend the concept of global approach of the MAPS to local MAPS which allows us to alleviate the difficulty of ill-conditioning and finding a suitable shape parameter. In Section 4, to validate our proposed approach, we compare our proposed method to the LMQ through three numerical examples.

2. The method of approximate particular solutions (MAPS)

Before we introduce the proposed local scheme we briefly review the formulation of the method of approximate particular solutions [1]. To illustrate the basic idea, we consider the following elliptic partial differential equation in a bounded domain Ω with its boundary $\partial\Omega$ in 2D, $\mathbf{x} = (x, y) \in \Omega$,

$$\Delta u + a(\mathbf{x})u_x + b(\mathbf{x})u_y + c(\mathbf{x})u = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{1}$$

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

where Δ is the Laplacian, $a(\mathbf{x})$, $b(\mathbf{x})$, $c(\mathbf{x})$, $f(\mathbf{x})$, and $g(\mathbf{x})$ are given functions. $\mathcal{B}$ is a boundary differential operator.

The basic idea of the MAPS is to rearrange (1) into the following Poisson-type equation

$$\Delta u(\mathbf{x}) = H(\mathbf{x}, u, u_x, u_y), \quad \boldsymbol{x} \in \Omega, \tag{3}$$

where

$$H(\mathbf{x}, u, u_x, u_y) = -a(\mathbf{x})u_x - b(\mathbf{x})u_y - c(\mathbf{x})u + f(\mathbf{x}). \tag{4}$$

H can be approximated by ϕ

$$H(\mathbf{x}, u, u_x, u_y) = \sum_{i=1}^{N} \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|), \quad \mathbf{x}, \mathbf{x}_i \in \Omega, \tag{5}$$

where $\|\cdot\|$ is the Euclidean norm, $\{\mathbf{x}_i\}_1^N$ are the interpolation points, and $\phi : \mathbb{R}_+ \to \mathbb{R}$ is a univariate function.

Using radial basis functions (RBFs), an approximate particular solution to (3) is given by

$$\hat{u}_p(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \Phi(\|\mathbf{x} - \mathbf{x}_i\|), \tag{6}$$

where on introducing $r = \|\mathbf{x} - \mathbf{x}_i\|$, Φ is obtained by analytically solving

$$\Delta\Phi(r) = \phi(r). \tag{7}$$

Note that $\Delta = 1/r(\mathrm{d}/\mathrm{d}r(r(\mathrm{d}/\mathrm{d}r)))$ in 2D. Φ in (7) can be obtained by repeated integration of ϕ [5,24]. Four commonly used RBFs, ϕ, and their corresponding particular solutions, Φ, are shown in Table 1.

Table 1
Radial basis functions and their corresponding particular solutions.

Type of RBF	$\phi(r)$	$\Phi(r)$
MQ	$\sqrt{r^2+c^2}$	$\frac{1}{9}(4c^2+r^2)\sqrt{r^2+c^2}-\frac{c^3}{3}\ln(c+\sqrt{r^2+c^2})$
IMQ	$\frac{1}{\sqrt{r^2+c^2}}$	$\sqrt{r^2+c^2}-c\ln(c+\sqrt{r^2+c^2})-c\ln 2c$
Polyharmonic spline	$r^{2m}\ln r$	$\frac{r^{2m+2}\ln r}{4(m+1)^2}-\frac{r^{2m+2}}{4(m+1)^3}$
Polyharmonic spline	r^{2m-1}	$\frac{r^{2m+1}}{(2m+1)^2}$

Let us assume the solution of (1)–(2) can be approximated by

$$u(\mathbf{x}) \simeq \hat{u}(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \Phi(\|\mathbf{x}-\mathbf{x}_i\|). \tag{8}$$

Then from (7) we have

$$\Delta u \simeq \Delta\hat{u} = \sum_{i=1}^{N} \alpha_i \Delta\Phi(\|\mathbf{x}-\mathbf{x}_i\|) = \sum_{i=1}^{N} \alpha_i \phi(\|\mathbf{x}-\mathbf{x}_i\|). \tag{9}$$

From (5) and (9), we have

$$\sum_{i=1}^{N} \alpha_i \phi(\|\mathbf{x}-\mathbf{x}_i\|) = -a(\mathbf{x})\hat{u}_x - b(\mathbf{x})\hat{u}_y - c(\mathbf{x})\hat{u} + f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{10}$$

and

$$u_x \simeq \hat{u}_x = \sum_{i=1}^{N} \alpha_i \Phi_x(\|\mathbf{x}-\mathbf{x}_i\|), \tag{11}$$

$$u_y \simeq \hat{u}_y = \sum_{i=1}^{N} \alpha_i \Phi_y(\|\mathbf{x}-\mathbf{x}_i\|). \tag{12}$$

We can reformulate (10) as

$$\sum_{i=1}^{N} \alpha_i \Theta(\|\mathbf{x}-\mathbf{x}_i\|) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{13}$$

where

$$\Theta(\|\mathbf{x}-\mathbf{x}_i\|) = \phi(\|\mathbf{x}-\mathbf{x}_i\|) + a(\mathbf{x})\Phi_x(\|\mathbf{x}-\mathbf{x}_i\|) + b(\mathbf{x})\Phi_y(\|\mathbf{x}-\mathbf{x}_i\|) + c(\mathbf{x})\Phi(\|\mathbf{x}-\mathbf{x}_i\|).$$

The boundary condition in (2) becomes

$$\sum_{i=1}^{N} \alpha_i \mathcal{B}\Phi(\|\mathbf{x}-\mathbf{x}_i\|) = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \tag{14}$$

By the use of the collocation method, $\{\alpha_i\}_{i=1}^{N}$ can be obtained through (13) and (14).

3. Localized method of approximate particular solutions (LMAPS)

To reduce the size of the dense matrices arising from the global scheme, we introduce a local scheme for the MAPS that was discussed in Section 2.

Let $\{\mathbf{x}_i\}_1^N$ be a set of collocation points in $\Omega \cup \partial\Omega$. For each $\mathbf{x}_i \in \Omega$ we choose n nearest neighbor points (including $\mathbf{x}_i$ itself) $\Omega_i = \{\mathbf{x}_k^i\}_{k=1}^{n}$, in which $x_k^i \equiv x_{k(i)}$, denotes the local indexing for each collocation point associated or belonging to Ω_i. The construction requires that $\Omega_i \cap \Omega_j \neq \emptyset$ for some $j \neq i$, and $\{\mathbf{x}_i\}_1^N = \cup_i \Omega_i$. In this section our purpose is to formulate a numerical scheme to approximate $u(\mathbf{x})$ and its derivatives at all the collocation points $\{\mathbf{x}_i\}_1^N$. Since these points can be selected arbitrarily in the domain, we can always choose the points where the approximate solutions are needed as the collocation points.

Consider the collocation method on the local domain Ω_i, and let $\mathbf{x}_i = \mathbf{x}_j^i \in \Omega_i$ for some $j \leq n$. Then $u(\mathbf{x}_i)$ can be approximated as shown in (8) which is the following

$$u(\mathbf{x}_i) \simeq \hat{u}(\mathbf{x}_i) = \sum_{k=1}^{n} \alpha_k \Phi(\|\mathbf{x}_i-\mathbf{x}_k^i\|). \tag{15}$$

Assume that within each Ω_i, using the collocation method, we obtain the resulting linear system

$$\begin{bmatrix} \hat{u}(\mathbf{x}_1^i) \\ \hat{u}(\mathbf{x}_2^i) \\ \vdots \\ \hat{u}(\mathbf{x}_n^i) \end{bmatrix} = \begin{bmatrix} \Phi(\|\mathbf{x}_1^i - \mathbf{x}_1^i\|) & \Phi(\|\mathbf{x}_1^i - \mathbf{x}_2^i\|) & \dots & \Phi(\|\mathbf{x}_1^i - \mathbf{x}_n^i\|) \\ \Phi(\|\mathbf{x}_2^i - \mathbf{x}_1^i\|) & \Phi(\|\mathbf{x}_2^i - \mathbf{x}_2^i\|) & \dots & \Phi(\|\mathbf{x}_2^i - \mathbf{x}_n^i\|) \\ \vdots & \vdots & \vdots & \vdots \\ \Phi(\|\mathbf{x}_n^i - \mathbf{x}_1^i\|) & \Phi(\|\mathbf{x}_n^i - \mathbf{x}_2^i\|) & \dots & \Phi(\|\mathbf{x}_n^i - \mathbf{x}_n^i\|) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}. \tag{16}$$

Denote the matrix on the right-hand side of (16) as $\boldsymbol{\Phi}_n$.

It can be proved [25] that $\boldsymbol{\Phi}_n$ is non-singular such that the inverse matrix can always be computed provided that all the nodal points inside Ω_i are distinct points. The unknown coefficients in (16) can be written as follows

$$\boldsymbol{\alpha} = \boldsymbol{\Phi}_n^{-1}\hat{\mathbf{u}}_n, \tag{17}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$, $\hat{\mathbf{u}}_n = [\hat{u}(\mathbf{x}_1^i), \hat{u}(\mathbf{x}_2^i), \dots, \hat{u}(\mathbf{x}_n^i)]^T$. Hence, $\hat{u}(\mathbf{x}_i)$ in (15) can be expressed in terms of the function values at n nodal points, $\hat{\mathbf{u}}_n$, i.e.

$$\begin{aligned} \hat{u}(\mathbf{x}_i) &= \sum_{k=1}^{n} \alpha_k \Phi(\|\mathbf{x}_i - \mathbf{x}_k^i\|) = \hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\alpha} = \hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}\hat{\mathbf{u}}_n \\ &= \boldsymbol{\Psi}_n(\mathbf{x}_i)\hat{\mathbf{u}}_n, \end{aligned} \tag{18}$$

where

$$\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i) = [\Phi(\|\mathbf{x}_i - \mathbf{x}_1^i\|), \Phi(\|\mathbf{x}_i - \mathbf{x}_2^i\|), \dots, \Phi(\|\mathbf{x}_i - \mathbf{x}_n^i\|)], \tag{19}$$

and

$$\boldsymbol{\Psi}_n(\mathbf{x}_i) = \hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1} = [\varphi_1, \varphi_2, \dots, \varphi_n]. \tag{20}$$

In the local approach, the formulation of (18) is preferred to (16). In (18) $\hat{u}(\mathbf{x}_i)$ is expressed in terms of the function values of u at the n local nodal points.

Let

$$\hat{\mathbf{u}}_N = [\hat{u}(\mathbf{x}_1), \hat{u}(\mathbf{x}_2), \dots, \hat{u}(\mathbf{x}_N)]^T. \tag{21}$$

We try to reformulate (18) in terms of global $\hat{\mathbf{u}}_N$ instead of local $\hat{\mathbf{u}}_n$. This can be done by padding the vector $\boldsymbol{\Psi}_n(\mathbf{x})$ with zero entries based on the mapping between $\hat{\mathbf{u}}_n$ and $\hat{\mathbf{u}}_N$. It follows that

$$\hat{u}(\mathbf{x}_i) = \boldsymbol{\Psi}_N(\mathbf{x}_i)\hat{\mathbf{u}}_N \tag{22}$$

is equivalent to (18), where $\boldsymbol{\Psi}_N(\mathbf{x}_i)$ is a vector with N components that is obtained by inserting $N - n$ zeros into $\boldsymbol{\Psi}_n(\mathbf{x}_i)$ at the proper places.

For example, let us assume $N = 100$, $n = 3$, and $\Omega_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \mathbf{x}_3^i\} = \{\mathbf{x}_{20}, \mathbf{x}_{23}, \mathbf{x}_{27}\}$. Then, we insert 97 zeros into the n-vector, $\boldsymbol{\Psi}_n(\mathbf{x}_i)$, given in (20), thereby padding the vector at positions other than 20, 23, and 27, as shown explicitly in

$$\boldsymbol{\Psi}_N(\mathbf{x}_i) = [0, 0, \dots, \underbrace{\varphi_1}_{\text{20th}}, 0, 0, \underbrace{\varphi_2}_{\text{23th}}, 0, 0, 0, \underbrace{\varphi_3}_{\text{27th}}, 0, \dots, \underbrace{0}_{\text{100th}}]. \tag{23}$$

In (23) there are 19 zeros before φ_1 and 73 zeros after φ_3. This zero padding keeps track of the original position at each local point so that $\boldsymbol{\Psi}_N(\mathbf{x}_i)$ can be easily obtained from $\boldsymbol{\Psi}_n(\mathbf{x}_i)$. The procedure is much the same as the process of matrix assembly in other local methods.

Next, we observe

$$\begin{aligned} \Delta\hat{u}(\mathbf{x}_i) &= \sum_{k=1}^{n} \alpha_k \Delta\Phi(\|\mathbf{x}_i - \mathbf{x}_k^i\|) \\ &= \Delta\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\alpha} \\ &= \Delta\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}\hat{\mathbf{u}}_n \\ &= \boldsymbol{\Lambda}_n(\mathbf{x}_i)\hat{\mathbf{u}}_n \\ &= \boldsymbol{\Lambda}_N(\mathbf{x}_i)\hat{\mathbf{u}}_N, \end{aligned} \tag{24}$$

where

$$\boldsymbol{\Lambda}_n(\mathbf{x}_i) = \Delta\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}, \tag{25}$$

and $\boldsymbol{\Lambda}_N(\mathbf{x}_i)$ is the expansion of $\boldsymbol{\Lambda}_n(\mathbf{x}_i)$ obtained by adding zero entries as mentioned above. Note that the Laplacian in (25) is applied to the radial basis functions that are the components of the matrix $\boldsymbol{\Phi}_n$. Since (18) and (24) hold for any $\{\mathbf{x}_i\}_1^N$, we have a $N \times N$ sparse system of equations with N unknown $\{\hat{u}(\mathbf{x}_i)\}_1^N$ which are approximate values of $\{u(\mathbf{x}_i)\}_1^N$, respectively.

We continue by outlining the application of this local scheme to solve (1)–(2). Let $\{\mathbf{x}_i\}_1^{\mathscr{N}_i}$ be the interior points in the domain Ω and $\{\mathbf{x}_i\}_{\mathscr{N}_i+1}^{\mathscr{N}_i+\mathscr{N}_b}$ be the boundary points on $\partial\Omega$ and $N = \mathscr{N}_i + \mathscr{N}_b$. For each interior point $\mathbf{x}_i \in \Omega$, we choose Ω_i that contains n nearest neighbor points of $\mathbf{x}_i$. From (13), for $1 \le i \le \mathscr{N}_i$, we have

$$\begin{aligned} f(\mathbf{x}_i) &= \sum_{k=1}^{n} \alpha_k \Theta(\|\mathbf{x}_i - \mathbf{x}_k^i\|) \\ &= \hat{\boldsymbol{\Theta}}_n(\mathbf{x}_i)\boldsymbol{\alpha} \\ &= \hat{\boldsymbol{\Theta}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}\hat{\mathbf{u}}_n \\ &= \boldsymbol{\Xi}_n(\mathbf{x}_i)\hat{\mathbf{u}}_n, \end{aligned} \tag{26}$$

where $\hat{\boldsymbol{\Theta}}_n$ is defined in a similar fashion as $\hat{\boldsymbol{\Phi}}_n$ in (19), $\boldsymbol{\alpha} = \boldsymbol{\Phi}_n^{-1}\hat{\mathbf{u}}_n$ is given in (17), and $\boldsymbol{\Xi}_n(\mathbf{x}_i) = \hat{\boldsymbol{\Theta}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}$. Similar to (18) and (22), we can extend $\hat{\mathbf{u}}_n$ to $\hat{\mathbf{u}}_N$ in (26) as

$$f(\mathbf{x}_i) = \boldsymbol{\Xi}_N(\mathbf{x}_i)\hat{\mathbf{u}}_N, \quad 1 \le i \le \mathscr{N}_i, \tag{27}$$

where $\boldsymbol{\Xi}_N(\mathbf{x}_i)$ is the extension of $\boldsymbol{\Xi}_n(\mathbf{x}_i)$ obtained by padding the vector with $N - n$ zeros in the proper position indicated above. Furthermore, from (14), for $\mathscr{N}_i + 1 \le i \le N$, we have

$$\begin{aligned} g(\mathbf{x}_i) &= \sum_{k=1}^{n} \alpha_k \mathscr{B}\Phi(\|\mathbf{x}_i - \mathbf{x}_k^i\|) \\ &= \mathscr{B}\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\alpha} \\ &= \mathscr{B}\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}\hat{\mathbf{u}}_n, \end{aligned} \tag{28}$$

where $\mathscr{B}\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)$ is defined in a similar fashion as $\Delta\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)$ in (24). Similarly, we can extend (28) from $\hat{\mathbf{u}}_n$ to $\hat{\mathbf{u}}_N$ as before. Then, we have

$$g(\mathbf{x}_i) = \boldsymbol{\Upsilon}_N(\mathbf{x}_i)\hat{\mathbf{u}}_N, \quad \mathscr{N}_i + 1 \le i \le N, \tag{29}$$

where $\boldsymbol{\Upsilon}_N(\mathbf{x}_i)$ is the extension of $\mathscr{B}\hat{\boldsymbol{\Phi}}_n(\mathbf{x}_i)\boldsymbol{\Phi}_n^{-1}$ obtained by inserting $N - n$ zeros. From (27) and (29), we have the following sparse system of equations

$$\begin{bmatrix} \boldsymbol{\Xi}_N(\mathbf{x}_1) \\ \vdots \\ \boldsymbol{\Xi}_N(\mathbf{x}_{\mathscr{N}_i}) \\ \boldsymbol{\Upsilon}_N(\mathbf{x}_{\mathscr{N}_i+1}) \\ \vdots \\ \boldsymbol{\Upsilon}_N(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}(\mathbf{x}_1) \\ \vdots \\ \hat{\mathbf{u}}(\mathbf{x}_{\mathscr{N}_i}) \\ \hat{\mathbf{u}}(\mathbf{x}_{\mathscr{N}_i+1}) \\ \vdots \\ \hat{\mathbf{u}}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_{\mathscr{N}_i}) \\ g(\mathbf{x}_{\mathscr{N}_i+1}) \\ \vdots \\ g(\mathbf{x}_N) \end{bmatrix}. \tag{30}$$

By solving this sparse system of equations, we obtain the approximate solution to u at all given nodes.

The availability of efficient sparse matrix solvers changes the numerical solution of the problem from one with a dense, ill-conditioned linear system, into one that contains a sparse matrix. Furthermore, the availability of a wide range of efficient sparse matrix solvers makes this approach much more amenable to solving large-scale problems in engineering and the applied sciences.

4. Numerical results

In the numerical implementation of the local meshless method, it is important to identify the nearest n neighboring points of each computational node. For large numbers of interpolation points, the efficiency of the search algorithm is an important consideration. Among these algorithms, the kd-tree is very efficient, using a space-partitioning data structure for arranging points in a k-dimensional space. The construction of the kd-tree is needed before the search of the nearest neighbor points, however the search can be done efficiently using the properties of the tree to quickly eliminate large portions of the search space. Finding the nearest point is an $O(\log N)$ operation in the case of N randomly distributed points.[1] In this section, we adopt the kd-tree search algorithm to find the nearest n neighbor points.

The effectiveness of our numerical scheme can be assessed by comparing numerical results with analytical solutions as the number of local nodes is varied. The high quality results attainable through the global MAPS are not examined, and in general, the use of the local methods results in some slight decrease in accuracy.

[1] For the details of the kd-tree search algorithm, we refer readers to Ref. [26]. The computer code for the construction of kd-tree is available.

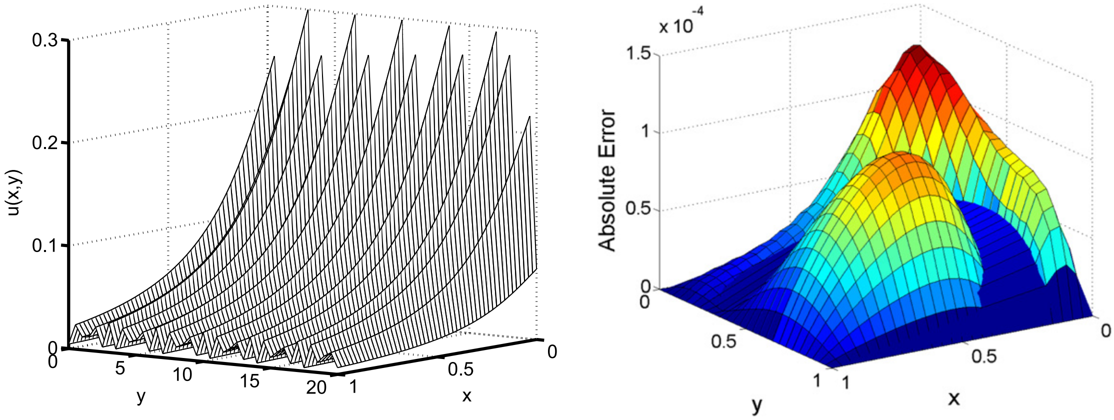

Fig. 1. Left: profile of the solution to the PDE in Example 1 on the domain $[0, 1] \times [0, 20]$. Right: the absolute errors of LMAPS in Example 1 with $L = 1, n = 7, \mathcal{S}_n = 25, c = 9.3$.

Estimates of the numerical accuracy are based on the root mean squared error (RMSE) and the root mean squared error of the spatial derivative with respect to x (RMSEx),

$$\varepsilon = \sqrt{\frac{1}{\mathcal{N}_t}\sum_{i=1}^{\mathcal{N}_t}(\hat{u}(\mathbf{x}_i) - u(\mathbf{x}_i))^2}, \tag{31}$$

$$\varepsilon_x = \sqrt{\frac{1}{\mathcal{N}_t}\sum_{i=1}^{\mathcal{N}_t}(\hat{u}_x(\mathbf{x}_i) - u_x(\mathbf{x}_i))^2}, \tag{32}$$

respectively, and where $\mathcal{N}_t$ is the number of test points.

To further validate our proposed numerical algorithm, we make comparison of our LMAPS with LMQ [10]. In addition, to show the effect of shape parameter c of MQ and IMQ in local methods, we test c from 0.1 to 10.0. In contrast to the global methods, we found the shape parameter c is insensitive using local approaches.

We denote by $\mathcal{N}_i$ the number of interior points, $\mathcal{N}_b$ the number of boundary points, with $N = \mathcal{N}_i + \mathcal{N}_b$, n the number of nearest neighbor points, and $\mathcal{S}_n$ the number of partition points in $(0, 1)$. The optimal shape parameter of MQ or IMQ is denoted as c_{opt}.

Example 1. Consider the following Poisson problem

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

where $\Omega \cup \partial\Omega = \{(x, y) : 0 \le x \le 1, 0 \le y \le L\}$. We note that the known functions f and g are given according to the following exact solution

$$u(x, y) = \frac{1.25 + \cos(5.4y + 2.7)}{6(1 + (3x + 0.5)^2)}.$$

The computational domain is a rectangular $[0, 1] \times [0, L]$. We use MQ as a basis function to solve the above problem for various L. The nodes are distributed uniformly. Let $\mathcal{N}_i = \mathcal{S}_n^2 L + \mathcal{S}_n(L - 1)$ and $\mathcal{N}_b = 2(L + 1)(\mathcal{S}_n + 1)$. We compute c_{opt} and the corresponding ε and ε_x on the N interior and boundary points. The profile of exact solution is shown on the left of Fig. 1, and clearly shows that, while the solution is smooth, it is rapidly varying. This is an important consideration in assessing the effectiveness of LMAPS in accurately capturing the behavior of the solution while only using a limited number of local nodes to construct the solution. The right figure of Fig. 1 shows the absolute errors on the domain $[0, 1] \times [0, 1]$, where $n = 7, \mathcal{S}_n = 25, c = 9.3$ are used. The estimates near the boundary as in any other region, are dependent on the construction of the local domains. The maximum errors appear in the region where the analytical solution has relative large values.

In Table 2, we compute the ε and ε_x errors using various nearest neighbor points n with $L = 20$. As n increases, both ε and ε_x are expected to improve while the computational efficiency will decrease. In this example, LMAPS and LMQ have similar accuracy using optimal c. In Table 3, the optimal shape parameter is stable and does not depend on the value of L. This is mainly because we have the same point distribution in each square $[0, 1] \times [i, i + 1]$, i.e., in Table 3, we always choose $30 \times 30 = 900$ inside $[0, 1] \times [i, i + 1]$. As we have observed, we can manage as many as 900,000 interpolation points and obtain good accuracy without problem. Table 3 shows ε and ε_x for various L.

Table 2
The ε and ε_x with various n using $L = 20$, $\mathcal{S}_n = 25$.

n	LMAPS			LMQ		
	ε	ε_x	c_{opt}	ε	ε_x	c_{opt}
7	9.46×10^{-5}	4.78×10^{-3}	9.3	5.87×10^{-5}	3.83×10^{-3}	0.8
9	5.88×10^{-5}	2.20×10^{-3}	5.3	8.34×10^{-5}	1.81×10^{-3}	0.5
11	8.99×10^{-5}	1.93×10^{-3}	2.8	8.29×10^{-5}	1.64×10^{-3}	0.4

Table 3
The ε and ε_x using different N for $n = 9$, $\mathcal{S}_n = 30$.

N	LMAPS			LMQ			
	ε	ε_x	c_{opt}	ε	ε_x	c_{opt}	L
99,232	1.10×10^{-4}	1.52×10^{-3}	1.6	5.96×10^{-5}	1.25×10^{-3}	0.5	100
198,432	1.08×10^{-4}	1.52×10^{-3}	1.6	5.97×10^{-5}	1.24×10^{-3}	0.5	200
376,992	1.11×10^{-4}	1.53×10^{-3}	1.6	5.99×10^{-5}	1.24×10^{-3}	0.5	380
496,032	1.11×10^{-4}	1.54×10^{-3}	1.6	5.98×10^{-5}	1.24×10^{-3}	0.5	500
595,232	1.11×10^{-4}	1.52×10^{-3}	1.6	5.99×10^{-5}	1.24×10^{-3}	0.5	600
694,432	1.10×10^{-4}	1.52×10^{-3}	1.6	5.98×10^{-5}	1.24×10^{-3}	0.5	700
803,552	1.09×10^{-4}	1.52×10^{-3}	1.6	5.97×10^{-5}	1.24×10^{-3}	0.5	810
922,592	1.11×10^{-4}	1.53×10^{-3}	1.6	5.98×10^{-5}	1.24×10^{-3}	0.5	930

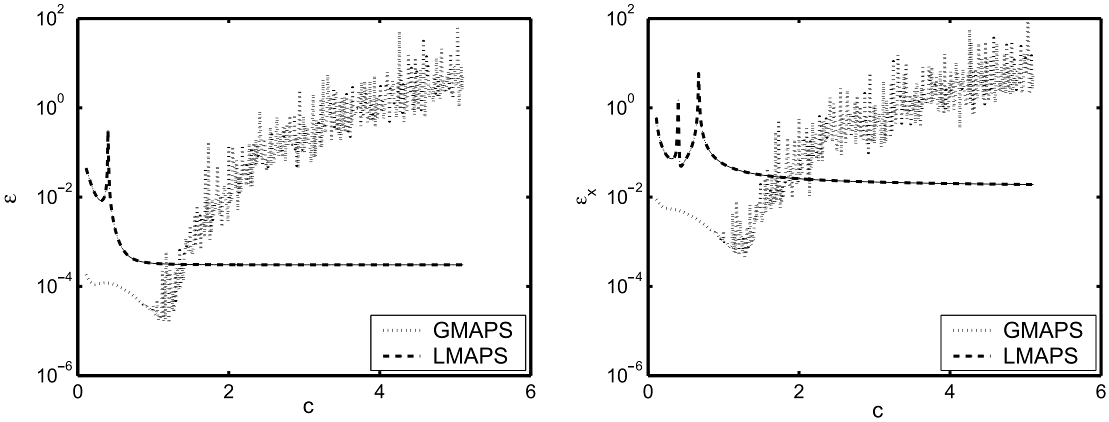

Fig. 2. The profiles of the ε and ε_x w.r.t. shape parameter c for Example 1 using LMAPS and GMAPS with $L = 1$, $\mathcal{S}_n = 10$, $n = 7$.

In Fig. 2, the shape parameter c is more stable and easy to predict using the LMAPS than global approach. It is known that the determination of optimal shape parameter is still an outstanding research problem [22,23]. Furthermore, we also show we can achieve both high accuracy and manage to handle a large number of interpolation points using local methods.

The left figure in Fig. 3 shows the rate of convergence of LMAPS with various numbers of nearest neighbor points n on the domain $[0, 1] \times [0, 1]$, where $\mathcal{S}_n = 10$ is considered. As we have seen from Table 2, the accuracy of the proposed method improves as n increases, but both RMSE and RMSEx become stable very quickly. The right figure in Fig. 3 shows the rate of convergence of LMAPS with various numbers of collocation points N, where $h = 1/(\mathcal{S}_n+1)$, $N = 4(\mathcal{S}_n+1)+\mathcal{S}_n^2 = (\mathcal{S}_n+2)^2$. As h decreased, the accuracy is increased as we expect.

Example 2. We consider the following modified Helmholtz equation

$$(\Delta - 100)u(x, y) = f(x, y), \quad (x, y) \in \Omega, \tag{33}$$

$$u(x, y) = g(x, y), \quad (x, y) \in \partial\Omega, \tag{34}$$

where f and g are chosen according to the following exact solution

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

The computational domain is bounded by the curve defined by the following parametric equation:

$$\partial\Omega = \{(x, y) \mid x = \rho\cos\theta, y = \rho\sin\theta, 0 \le \theta \le 2\pi\}, \tag{35}$$

where

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

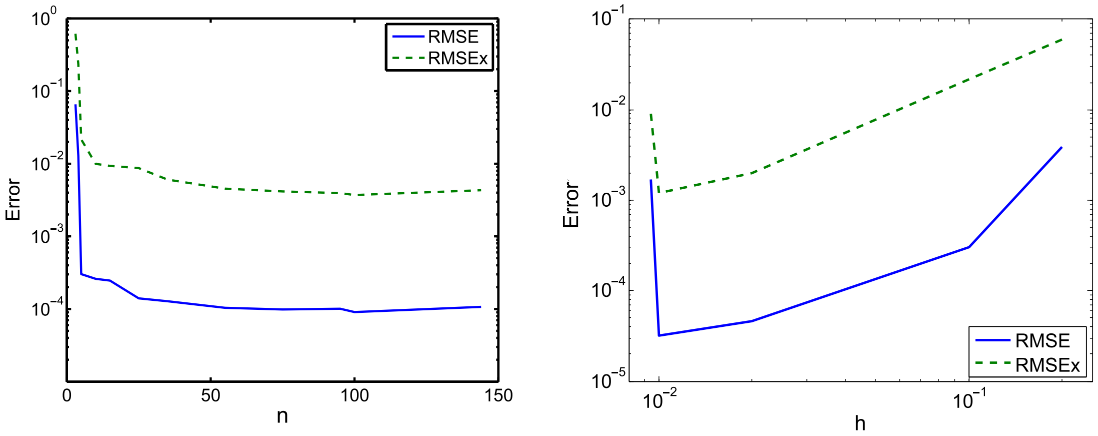

Fig. 3. Left: the rate of convergence of LMAPS with various numbers of nearest neighbor points n in Example 1 on the domain $[0, 1] \times [0, 1]$, $\mathcal{S}_n = 10$. Right: the rate of convergence of LMAPS with various numbers of collocation points N in Example 1 on the domain $[0, 1] \times [0, 1]$, where $n = 5, h = 1/(\mathcal{S}_n + 1), N = (\mathcal{S}_n + 2)^2$.

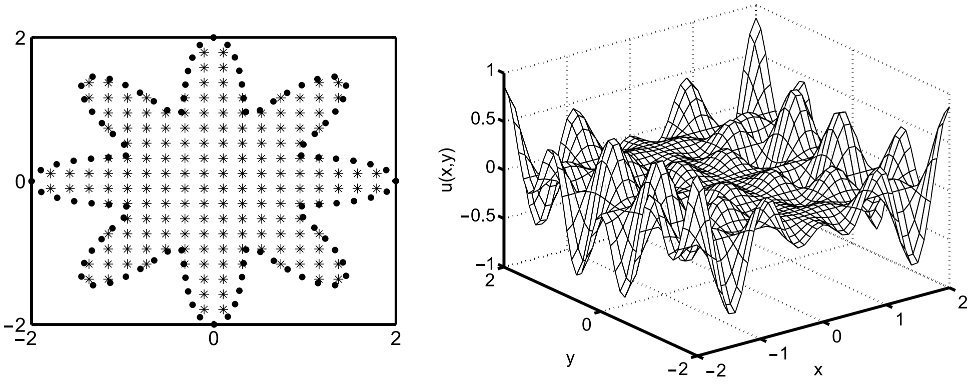

Fig. 4. The profile of the domain and the distribution of interior and boundary points (left), and the profile of the solution to the PDE in Example 2 in the extended domain (right).

The profiles of computational domain and the exact solution can be found in Fig. 4. This problem is designed to demonstrate the utility of LMAPS in solving problems on irregular domains, and as in Example 1, to examine a case where the solution is again rapidly oscillating on the domain.

We use the LMAPS and LMQ with MQ and IMQ as basis functions to solve the above problem. Let $\mathcal{N}_t = 172$ be the number of test points. For simplicity, we can simply add the test points as part of the interior points. Hence, in this example we have $N = \mathcal{N}_i + \mathcal{N}_b + \mathcal{N}_t$. Table 4 shows that both ε and ε_x improve when N becomes larger using LMAPS and LMQ.

In Tables 5 and 6, we compute the ε and ε_x errors using LMAPS and LMQ with MQ and IMQ, respectively. The numerical results show that the IMQ has slightly better results. In Fig. 5 we notice that LMAPS is clearly more stable and accurate than LMQ in terms of shape parameter c. It is difficult to identify the optimal shape parameter c_{opt} using LMQ. Using LMAPS in this example, finding c_{opt} is not a critical issue.

This final point deserves some attention. In having moved to a local scheme, we have gained an important advantage beyond the design objective of improving the solvability of the linear system. In regard to the determination of the optimum for the values of c, i.e., c_{opt}, we have a much simpler problem to solve. Since the matrices are much smaller, provided that the spacing of the points in each local domain is approximately the same, the value of c that optimizes the solution of the problem can also be expected to be the same. In effect we have regularized the construction of c_{opt}.

The result for c_{opt} should also hold provided that the shape of the local domains is not excessively stretched or distorted. This would hardly be surprising, in as much as all the local methods seem to share a need to keep the mesh or grid from being excessively distorted to minimize errors.

Example 3. We consider a more general type of partial differential equation

$$\Delta u(x, y) + y\cos(y)u_x(x, y) + \sinh(x)u_y(x, y) + 10xyu(x, y) = f(x, y), \quad (x, y) \in \Omega,$$
$$\frac{\partial u(x, y)}{\partial \mathbf{n}} = g(x, y), \quad (x, y) \in \partial\Omega,$$

Table 4
The ε and ε_x for different N using MQ and $n = 9$.

$\mathcal{N}_i$	$\mathcal{N}_b$	LMAPS			LMQ		
		ε	ε_x	c_{opt}	ε	ε_x	c_{opt}
384	80	3.07×10^{-3}	8.34×10^{-2}	0.4	2.22×10^{-3}	5.31×10^{-2}	0.5
6600	300	7.14×10^{-5}	1.67×10^{-3}	1.7	6.19×10^{-5}	1.01×10^{-3}	1.4
20161	900	5.02×10^{-5}	8.91×10^{-4}	0.6	2.57×10^{-5}	3.68×10^{-4}	0.8

Table 5
The ε and ε_x with different values of n using MQ and $\mathcal{N}_i = 6,600$, $\mathcal{N}_b = 300$.

n	LMAPS			LMQ		
	ε	ε_x	c_{opt}	ε	ε_x	c_{opt}
7	1.24×10^{-4}	4.53×10^{-3}	3.6	1.25×10^{-4}	3.56×10^{-3}	0.4
9	3.01×10^{-4}	8.73×10^{-3}	1.7	1.88×10^{-4}	5.26×10^{-3}	1.7
11	7.14×10^{-5}	1.67×10^{-3}	1.7	6.19×10^{-5}	1.01×10^{-3}	1.4
13	3.02×10^{-5}	1.70×10^{-3}	0.7	1.48×10^{-5}	5.08×10^{-4}	0.6
15	9.81×10^{-6}	3.88×10^{-4}	0.6	3.32×10^{-6}	2.53×10^{-4}	0.6

Table 6
The ε and ε_x with different values of n using IMQ and $\mathcal{N}_i = 6{,}600$, $\mathcal{N}_b = 300$.

n	LMAPS			LIMQ		
	ε	ε_x	c_{opt}	ε	ε_x	c_{opt}
7	1.41×10^{-4}	4.32×10^{-3}	0.4	9.90×10^{-5}	3.37×10^{-3}	0.7
9	1.90×10^{-4}	5.31×10^{-3}	1.4	1.71×10^{-4}	4.79×10^{-3}	0.8
11	6.29×10^{-5}	8.46×10^{-4}	1.3	4.19×10^{-5}	7.47×10^{-4}	1.2
13	1.56×10^{-5}	5.91×10^{-4}	0.5	1.34×10^{-5}	5.15×10^{-4}	0.8
15	4.32×10^{-6}	3.31×10^{-4}	0.6	2.15×10^{-6}	1.39×10^{-4}	0.7

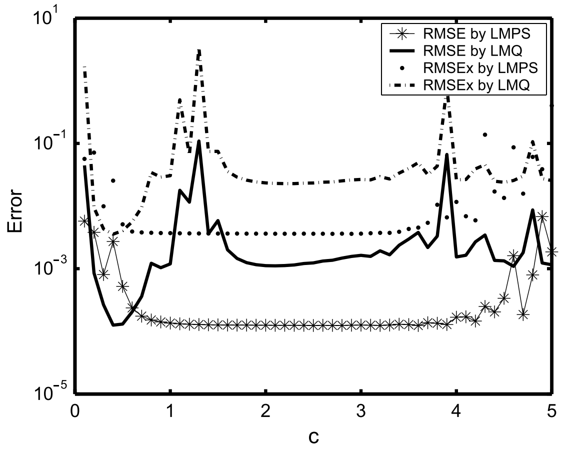

Fig. 5. The profiles of the ε and ε_x versus shape parameters c using LMAPS and LMQ with $\mathcal{N}_i = 6600$, $\mathcal{N}_b = 300$, $n = 7$ for Example 2.

where $\mathbf{n}$ is the outward normal derivative. The f and g are given according to the exact solution in Example 2. The domain is a unit square. The nodes are distributed uniformly. Let $\mathcal{N}_i = \mathcal{S}_n^2$ and $\mathcal{N}_b = 4(\mathcal{S}_n + 1)$. In this example, the MQ is chosen as the basis function.

In Tables 7 and 8, we observe that there are little improvements in accuracy for increasing n and $\mathcal{S}_n$ using LMAPS and LMQ. On the other hand, with increase of n or $\mathcal{S}_n$, the computational cost will increase.

The left figure of Fig. 6 shows the ε as a function of shape parameter in MQ RBF with different n. The optimal shape parameter becomes smaller when larger number of nearest points are used, the range of reasonable shape parameter also becomes smaller. The right figure of Fig. 6 shows the rate of convergence of LMAPS with various numbers of nearest neighbor points n, where $\mathcal{S}_n = 10$ is considered. The accuracy of the LMAPS is improving as n increasing, but both RMSE and RMSEx become stable very quickly. In Fig. 7 we show the rate of convergence of LMAPS and LMQ with respect to the mesh size h. Note that the performance of LMAPS and LMQ as shown in Fig. 7 are very close to each other.

Numerical studies show that implementation of the LMAPS shares certain observed behaviors with other local meshless approaches discussed in the literature. These include the behavior of the shape parameter associated with interpolation

Table 7
The ε and ε_x are obtained using different numbers of interpolation points and $\mathcal{S}_n = 30$.

n	LMAPS			LMQ		
	ε	ε_x	c_{opt}	ε	ε_x	c_{opt}
9	8.36×10^{-4}	3.48×10^{-3}	2.0	6.51×10^{-4}	3.43×10^{-3}	0.6
15	4.77×10^{-4}	2.05×10^{-3}	0.7	4.01×10^{-4}	1.49×10^{-3}	0.6
21	4.62×10^{-4}	1.38×10^{-3}	0.3	3.08×10^{-4}	1.09×10^{-3}	0.5
27	3.68×10^{-4}	1.03×10^{-3}	0.2	8.70×10^{-4}	1.48×10^{-3}	0.4
33	3.09×10^{-4}	9.74×10^{-4}	0.2	4.14×10^{-4}	1.20×10^{-3}	0.3

Table 8
The ε and ε_x using different $\mathcal{S}_n$ and $n = 9$.

$\mathcal{S}_n$	LMAPS			LMQ		
	ε	ε_x	c_{opt}	ε	ε_x	c_{opt}
10	6.81×10^{-3}	4.57×10^{-2}	0.8	9.07×10^{-3}	3.12×10^{-2}	1.3
25	8.25×10^{-4}	5.89×10^{-3}	2.9	1.02×10^{-3}	5.01×10^{-3}	0.6
40	4.82×10^{-4}	2.24×10^{-3}	1.2	3.32×10^{-4}	1.90×10^{-3}	0.6
55	3.09×10^{-4}	1.26×10^{-3}	0.9	1.63×10^{-4}	9.96×10^{-4}	0.6
70	2.16×10^{-4}	7.63×10^{-4}	0.6	1.00×10^{-4}	6.09×10^{-4}	0.6

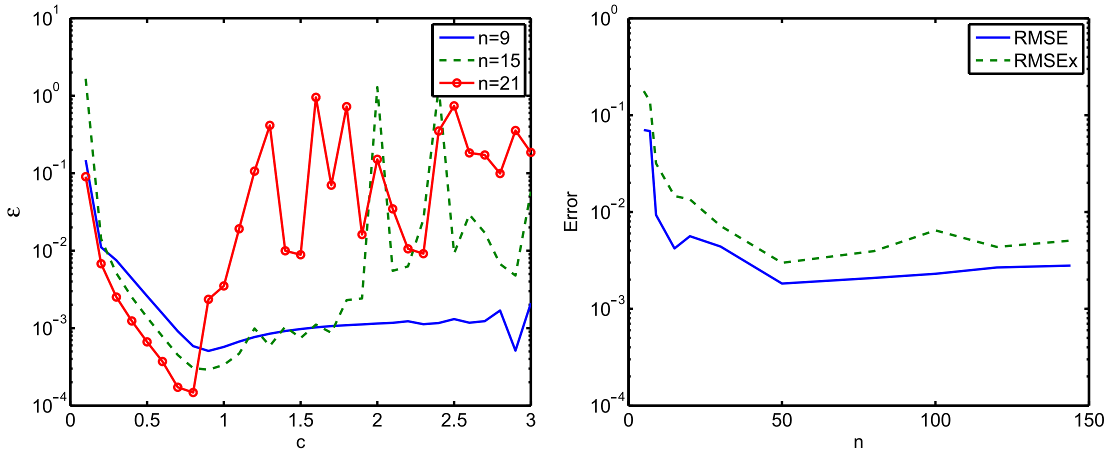

Fig. 6. Left: profiles of ε versus shape parameters c using LMAPS with different numbers of nearest points in local domains with $\mathcal{S}_n = 30$, $n = 9$, 15, and $n = 21$ for Example 3. Right: the rate of convergence of LMAPS with various numbers of nearest neighbor points n in Example 3, $\mathcal{S}_n = 10$ is used.

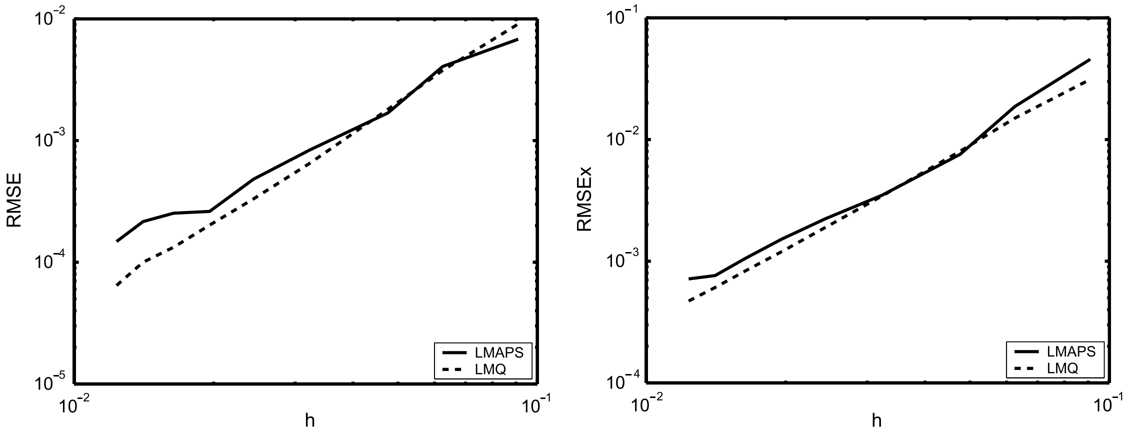

Fig. 7. Profiles of the optimal ε (left) and ε_x (right) using LMAPS (solid line) and LMQ (dotted line) using $h = 1/(\mathcal{S}_n + 1)$, $n = 9$ for Example 3.

and the rate of convergence of the local approach compared with the corresponding convergence of the global method. The LMAPS, however, is clearly more stable in terms of finding a suitable shape parameter c, and also yields a slightly more accurate approximation than LMQ.

5. Conclusions

For solving large-scale, realistic problems, the localized approaches is essential. Global methods are known to have many limitations regarding the computability of solutions, including ill-conditioning due to a dense matrix formulation, and difficulty in selecting optimal shape parameter if MQ RBFs are used.

The formulation of localized method offers an alternative to solve these larger problems without and of the limitations that are inherent in attempting to solve the fully coupled linear systems. Clearly, in these problems the accuracy of the numerical results were not in any manner unduly affected by localizing the computation, in much the same manner as finite difference or finite element methods do not. Clearly, the intrinsic nature of the elliptic and parabolic type PDEs considered in this study requires the coupling of the interior nodes to the boundary, i.e., avoiding the isolation of nodes, the analytical bounds on accuracy, stability and convergence remain to be worked out.

We have pushed the limit in solving large problems using nearly one million interpolation points with nothing more exceptional than a personal computer, obtaining excellent results in a reasonable amount of time. In the local approach, searching the local domain for each node is the most time consuming process. In Example 1, kd-tree uses 655.70 s to find and store all local domains for 922,592 interpolation nodes, and the LMAPS method uses 836.80 s to solve this nearly one million-node problem. The numerical performance of LMAPS and LMQ are similar; however, the LMAPS is more insensitive in terms of finding the suitable shape parameter of MQ or IMQ. In the local methods the size of matrix in the local linear system is relatively small, and does not vary much. This allows for an optimal choice of c compared with the large dense linear systems.

It is apparent that meshless methods using local schemes can compete with the traditional numerical methods for solving large-scale PDEs such as the finite element, finite difference or finite volume methods. It is fairly evident that the proposed MAPS can be easily extended to solving three-dimensional problems with irregular domains, and this is indeed where its strength may lie. As such, the approach offers the prospects of an efficient algorithm for solving more challenging problems in science and engineering.

Acknowledgement

The third author acknowledges the support of the Distinguished Overseas Visiting Scholar Fellowship funded by the Minister of Education in China and the support of NATO-RTA project under reference AVT-08/1.

References

[1] P.H. Wen, C.S. Chen, The method of particular solutions for solving scalar wave equations, The International Journal for Numerical Methods in Biomedical Engineering 26 (2010) 1878–1889.
[2] K.E. Atkinson, The numerical evaluation of particular solutions for Poisson's equation, IMA Journal of Numerical Analysis 5 (1985) 319–338.
[3] W. Chen, Symmetric boundary knot method, Engineering Analysis with Boundary Elements 26 (2002) 489–494.
[4] W. Chen, Y.C. Hon, Numerical convergence of boundary knot method in the analysis of Helmholtz, modified Helmholtz, and convection–diffusion problems, Computer Methods in Applied Mechanics and Engineering 192 (2003) 1859–1875.
[5] M.A. Golberg, C.S. Chen, The method of fundamental solutions for potential, Helmholtz and diffusion problems, in: M.A. Golberg (Ed.), Boundary Integral Methods: Numerical and Mathematical Aspects, WIT Press, 1998, pp. 103–176.
[6] C.S. Chen, C.M. Fan, P.H. Wen, The method of particular solutions for solving certain partial differential equations, Numerical Methods for Partial Differential Equations (2010) doi:10.1002/num.20631.
[7] John.T. Katsikadelis, The 2d elastostatic problem in inhomogeneous anisotropic bodies by the meshless analog equation method (maem), Engineering Analysis with Boundary Elements: Special Issue: BEM/MRM for Inhomogeneous Solids 32 (2008) 997–1005.
[8] N. Mai-Duy, T. Tran-Cong, Indirect RBFN method with thin plate splines for numerical solution of differential equations, Computer Modeling in Engineering & Science 4 (2003) 85–102.
[9] E. Divo, A.J. Kassab, An efficient localized rbf meshless method for fluid flow and conjugate hear transfer, ASME Journal of Heat Transfer 129 (2007) 124–136.
[10] C.K. Lee, X. Liu, S.C. Fan, Local multiquadric approximation for solving boundary value problems, Computational Mechanics 30 (2003) 396–409.
[11] B. Sarler, R. Vertnik, Meshfree explicit local radial basis function collocation method for diffusion problems, Computers and Mathematics with Applications 21 (2006) 1269–1282.
[12] C. Shu, H. Ding, K.S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two dimensional incompressible navier-stokeequations, Computer Methods and Applied Mechanics Engineering 192 (2003) 941–954.
[13] R. Vertnik, B. Sarler, Meshless local radial basis function collocation method for convective-diffusive solid–liquid phase change problems, International Journal of Numerical Methods for Heat and Fluid Flow 16 (2006) 617–640.
[14] E.J. Kansa, Y.C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations, Computcomputer & Mathematics. with Applications 39 (7–8) (2000) 123–137.
[15] Y.C. Hon, R. Schaback, X. Zhou, An adaptive greedy algorithm for solving large RBF collocation problems, Numerical Algorithms 32 (1) (2003) 13–25.
[16] C.S. Chen, M.A. Golberg, M. Ganesh, A.H.-D. Cheng, Multilevel compact radial functions based computational schemes for some elliptic problems, Computers and Mathematics with Application 43 (2002) 359–378.
[17] C.-S. Huang, Cheng-Feng Lee, A.H.-D. Cheng, Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method, Engineering Analysis with Boundary Elements 31 (7) (2007) 614–623.
[18] N.A. Libre, A. Emdadi, E.J. Kansa, M. Rahimian, M. Shekarchi, A stabilized rbf collocation scheme for neumann type boundary value problems, Computer Modeling in Engineering & Science 24 (2008) 61–80.
[19] A.H.-D. Cheng, D.-L. Young, J.-J. Tsai, Solution of Poisson's equation by iterative DRBEM using compactly supported, positive definite radial basis function, Engineering Analysis with Boundary Elements 24 (2000) 549–557.
[20] R.K. Beatson, L. Greengard, A short course on fast multipole methods, in: M. Ainsworth, J. Levesley, W. Light, M. Marletta (Eds.), Wavelets, Multilevel Methods and Elliptic PDEs, Oxford University Press, 1997, pp. 1–37.
[21] Leevan Ling, Robert Schaback, An improved subspace selection algorithm for meshless collocation methods, International Journal for Numerical Methods in Engineering 80 (2009) 1623–1639.

[22] Shmuel Rippa, An algorithm for selecting a good value for the parameter *c* in radial basis function interpolation, Advances in Computational Mathematics 11 (2–3) (1999) 193–210.

[23] J. Wertz, E.J. Kansa, L. Ling, The role of the multiquadric shape parameters in solving elliptic partial differential equations, Computer & Mathematical with Applications 51 (8) (2006) 1335–1348.

[24] A. Karageorghis, C.S. Chen, Y-S Smyrlis, Matrix decomposition rbf algorithm for solving 3d elliptic problems, Engineering Analysis with Boundary Elements 33 (2009) 1368–1373.

[25] J. Duchon, Splines minimizing rotation invariant semi-norms in Sobolev spaces: constructive theory of functions of several variables, in: W. Schempp, K. Zeller (Eds.), in: Lecture Notes in Mathematics, vol. 571, Springer-Verlag, Berlin, 1976, pp. 85–110.

[26] Steven Skiena, The Algorithm Design Manual, 2nd ed., Springer, 2008.