Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **10**, No. 4, pp. 1-16

DOI: 10.4208/aamm.OA-2017-0167
August 2018

On the Selection of a Good Shape Parameter of Localized Method of Approximated Particular Solutions

Hui Zheng[1], Guangming Yao[2], Lei-Hsin Kuo[3] and Xinxiang Li[4,*]

[1] *School of Civil Engineering and Architecture, Nanchang University, Nanchang 330031, China*
[2] *Department of Mathematics, Clarkson University, Potsdam, NY 13699-5815, USA*
[3] *Department of Mathematics and Statistics, University of West Florida, Pensacola, FL 32514, USA*
[4] *Department of Mathematics, Shanghai University, Shanghai 200444, China*

Received 26 June 2017; Accepted (in revised version) 19 November 2017

Abstract. In this paper, we propose a new approach for selecting suitable shape parameters of radial basis functions (RBFs) in the context of the localized method of approximated particular solutions. Traditionally, there ~~is~~ no direct connections on choosing good shape parameters and choosing interior and boundary nodes using the local collocation methods. As a result, the ~~approximation~~ of derivative functions ~~is~~ less accurate and the stability is also an issue. One of the focuses of this study is to select the interior and boundary nodes in a special way so that they are correlated. Furthermore, a test differential equation with known exact solution is selected and a good shape parameter for the given differential equation can be selected through a good shape parameter for the test differential equation. ~~Two~~ numerical examples, including a Poison's equation and an eigenvalue problem, are tested. Uniformly distributed node arrangement is compared with the proposed cross knot distribution ~~in Example 4.1~~ with Dirichlet boundary conditions and mixed boundary conditions. The numerical results show some potentials for the proposed node arrangements and shape parameter selections.

AMS subject classifications: ~~to be provided by authors~~
Key words: Method of ~~approximate~~ particular solutions, shape parameter, radial basis functions, ~~RBF~~ collocation methods, Kansa's method.

1 Introduction

The radial basis function (RBF) collocation method or the so-called Kansa's method [9] was proposed in early 1990's and has became very popular for solving various types

*Corresponding author.
Email: xinxiang.lee@t.shu.edu.cn (X. X. Li)

 ©2018 Global Science Press

GalleyProof

of problems in science and engineering. The main attractions of Kansa's method are its simplicity and high accuracy. Due to its simplicity, Kansa's method is especially useful for solving ~~high dimensional~~ problems with ~~complicate~~ domains. To alleviate the difficulty of dense and ill-conditioning system of linear equations in the formulation of the global RBF collocation methods, a number of localized RBF methods [13, 15, 17] were proposed for solving more challenging problems, which these methods can solve a system involves large number of RBF centers. As a result, the linear system created through collocation is sparse which allows us to solve large-scale problems in science and engineering. Despite all the favorable features of the newly developed RBF collocation methods, the accuracy of the approximated solutions heavily depends on the value of the shape parameters of RBFs. It is known that the determination of the optimal shape parameters of RBFs is still an outstanding research topic. There is still no theory or recipe for selecting the optimal shape parameters that can consistently apply to various applications. This issue has been studied by several authors such as Hardy [8], Franke [6], Foley [5], Carlson and Foley [1], Golberg et al. [7], Rippa [14], Kansa and Hon [10], Larson and Fornberg [12], to name just a few. Most of the proposed approaches were given through ~~experiment~~ or statistics. Each proposed technique has its advantages and drawbacks. In his paper, Rippa [14] believes that the shape parameter should depend on a number of factors such as the number of grid points, distribution of grid points, RBF functions, condition number and computer precision.

The purpose of this short paper is to propose another approach for choosing a good shape parameter for solving partial differential equations using localized RBF collocation methods. The proposed method for selecting a good shape parameter is suitable for many methods that involve RBF collocation. In particular, we implement the proposed approach in the context of the localized method of ~~approximate~~ particular solutions (LMAPS) [17]. In most of the RBF collocation methods, the number and the distribution of the interior and boundary points are selected in an arbitrary way and there is not close relationship between them. It is known that RBF collocation methods can produce accurate solution but less accurate for the corresponding derivative functions' approximations. To achieve a better accuracy, it is important to find a way to more accurately approximate the derivatives using RBF collocation methods.

This paper builds upon several observations. We first observe that a better approximation of derivative functions can be achieved if the boundary points and interior points are all uniformly lined up in each axis direction such as the point distribution used in the finite difference method. Next, for selecting the shape parameter, we propose to choose a test function which is a solution of a differential equation with the same differential operator as the given differential equation. As we shall see, a good shape parameter of the given differential equation can be chosen through the test function.

The structure of the paper is as follows. In Section 2, we give a brief review of the LMAPS. In Section 3, we propose a new approach to distribute the boundary and the interior nodes. In Section 4, ~~two~~ numerical examples are given to demonstrate the effectiveness of the proposed method. In Section 5, some concluding remarks are given.

Galley Proof

2 The localized method of approximate particular solutions (LMAPS)

In this section, we give a brief review of the LMAPS. Let L be a linear second-order elliptic partial differential operator, B be a boundary differential operator. We consider the following boundary value problem

$$Lu(x,y)=f(x,y), \qquad (x,y)\in\Omega, \tag{2.1a}$$
$$Bu(x,y)=g(x,y), \qquad (x,y)\in\partial\Omega, \tag{2.1b}$$

where Ω is a bounded and closed domain with a sufficiently smooth boundary $\partial\Omega$. We consider the case of the above boundary value problem has a unique solution.

Let $\{(x_j,y_j)\}_{j=1}^{n}$ be a set of interpolation points inside the domain Ω. For any point $(x_p,y_p)\in\Omega$, we create a local influence domain Ω_p, which is a region containing n_s neighboring interpolation points $\{(x_j,y_j)\}_{j=1}^{n_s}$ of (x_p,y_p). The method of particular solutions assumes the solution space is a finite vector space of a special kind of radial basis functions, which is so called particular solutions. The particular solutions are defined as the solution Φ to the following differential equations

$$L\Phi=\varphi, \tag{2.2}$$

in which φ is a commonly used radial basis function. Note that the particular solutions are derived analytically with respect to the given differential operator L and chosen radial basis function φ. More details can be found in [3]. By the method of particular solutions, $u(x_p,y_p)$ can be approximated by a linear combination of n_s radial basis functions in the following form:

$$u(x_p,y_p)\simeq\hat{u}(x_p,y_p)=\sum_{j=1}^{n_s}\alpha_j\Phi\left(\left\|(x_p,y_p)-(x_j,y_j)\right\|\right), \tag{2.3}$$

where $\{\alpha_j\}_{j=1}^{n_s}$ are coefficients to be determined, $\|\cdot\|$ is the Euclidean norm and $\Phi\left(\left\|(x,y)-(x_j,y_j)\right\|\right)$ are the n_s RBFs created by choosing n_s points in the local domain of (x_p,y_p), Ω_p. Since $\{(x_j,y_j)\}_{j=1}^{n_s}\subset\Omega_p$, Eq. (2.3) holds for every (x_j,y_j), $j=1,2,\cdots,n_s$. Thus, it follows that

$$\hat{\mathbf{u}}_{n_s}=\mathbf{\Phi}_{n_s}\boldsymbol{\alpha}_{n_s}, \tag{2.4}$$

where $\hat{\mathbf{u}}_{n_s}=[\hat{u}(x_1,y_1),\cdots,\hat{u}(x_{n_s},y_{n_s})]^T$ are unknown solution values to be approximated, $\boldsymbol{\alpha}_{n_s}=[\alpha_1,\alpha_2,\cdots,\alpha_{n_s}]^T$ are unknown coefficients to be determined and the collocation matrix in the local domain Ω_p is $\mathbf{\Phi}_{n_s}=\left[\Phi\left(\left\|(x_i,y_i)-(x_j,y_j)\right\|\right)\right]_{i,j=1}^{n_s}$. Rewrite Eq. (2.4), we have that

$$\boldsymbol{\alpha}_{n_s}=\mathbf{\Phi}_{n_s}^{-1}\hat{\mathbf{u}}_{n_s}. \tag{2.5}$$

Galley Proof

Note that the equation above is an expression, in which we will neither need the matrix inverse in numerical computation, nor it is reasonable to use practically. The expression is for elimination of the unknown coefficients $\boldsymbol{\alpha}$ so it is described by unknown approximations $\hat{\mathbf{u}}$. Plug the above expression in Eq. (2.3), we have that

$$\hat{u}(x_p,y_p)=\sum_{j=1}^{n_s}\alpha_j\Phi\left(\left\|(x_p,y_p)-(x_j,y_j)\right\|\right)=\boldsymbol{\Theta}_{n_s}\boldsymbol{\alpha}_{n_s}=\left(\boldsymbol{\Theta}_{n_s}\boldsymbol{\Phi}_{n_s}^{-1}\right)\hat{\mathbf{u}}_{n_s}, \tag{2.6}$$

where

$$\boldsymbol{\Theta}_{n_s}=\left[\Phi\left(\left\|(x_p,y_p)-(x_1,y_1)\right\|\right),\cdots,\Phi\left(\left\|(x_p,y_p)-(x_{n_s},y_{n_s})\right\|\right)\right]. \tag{2.7}$$

Thus, Eq. (2.1a) can be rewritten in the following form for any $(x_p,y_p)\in\Omega_p$

$$L\hat{u}(x_p,y_p)=\sum_{j=1}^{n_s}\alpha_j L\Phi\left(\left\|(x_p,y_p)-(x_j,y_j)\right\|\right)=\left(L\boldsymbol{\Theta}_{n_s}\boldsymbol{\Phi}_{n_s}^{-1}\right)\hat{\mathbf{u}}_{n_s}=f(x_p,y_p). \tag{2.8}$$

Note that the $L\boldsymbol{\Theta}_{n_s}\boldsymbol{\Phi}_{n_s}^{-1}$ can be viewed as solution to a linear system where the coefficients are given by $\boldsymbol{\Phi}_n$ and the right-hand-side of the system is given by $L\boldsymbol{\Theta}_{n_s}$. Thus, the matrix inverse is not actually involved, even though the expression exists. This is a similar technique that is described in [16]. It is easy to convert Eq. (2.8) into the global form

$$\left(L\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n=f\left(x_p,y_p\right),\quad (x_p,y_p)\in\Omega, \tag{2.9}$$

where $\hat{\mathbf{u}}_n=[\hat{u}(x_1,y_1),\hat{u}(x_2,y_2),\cdots,\hat{u}(x_n,y_n)]^T$ and $\left(L\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)$ is the mapping of $\left(L\boldsymbol{\Theta}_{n_s}\boldsymbol{\Phi}_{n_s}^{-1}\right)$ from local to global by inserting zeros in the proper position.

For $(x_p,y_p)\in\partial\Omega$, the formulation of the boundary condition is similar to the interior points. Hence, we have

$$\left(B\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n=g(x_p,y_p),\quad (x_p,y_p)\in\partial\Omega. \tag{2.10}$$

Let $\{(x_i,y_i)\}_{i=1}^{n_i}$ be the interior points, $\{(x_i,y_i)\}_{i=n_i+1}^{n_i+n_b}$ be the boundary points and $n=n_i+n_b$. By using Eqs. (2.9) and (2.10), the LMAPS discretize the original elliptic equation (2.1a)-(2.1b) into

$$\left(L\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n=f(x_j,y_j),\qquad j=1,2,\cdots,n_i, \tag{2.11a}$$

$$\left(B\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n=g(x_j,y_j),\qquad j=n_i+1,\cdots,n. \tag{2.11b}$$

For the Dirichlet boundary condition, Eq. (2.11b) becomes

$$\hat{u}(x_j,y_j)=g(x_j,y_j),\quad j=n_i+1,\cdots,n.$$

Notice that Eqs. (2.11a) and (2.11b) ~~are the~~ system of linear equations, with n unknowns which is the values of the approximated solution at nodal nodes. Additionally, each

Galley Proof

equation contains only n_s non-zero terms which obtained by considering the local influence domains of the nodal points. Thus, this is a sparse system of equations, which can be solved by sparse system solver such as MATLAB build-in solver. The solutions to the sparse system are the approximated solution of Eqs. (2.1a)-(2.1b) at all the nodal points.

In this paper the multiquadric (MQ), $\varphi=\sqrt{r^2+c^2}$, is used as the basis function. The corresponding particular solution, Φ in Eq. (2.2) for $L=\Delta$ in two-dimensional space is as follows [2,3]

$$\Phi(r)=\frac{1}{9}\left(4c^2+r^2\right)\sqrt{r^2+c^2}-\frac{c^3}{3}\ln\left(c+\sqrt{r^2+c^2}\right), \tag{2.12}$$

where r is the Euclidean distance.

3 Cross knot distribution (CKD) and test differential equation

In this section, a node distribution that aligns interior and boundary points in each axis is first proposed. Fig. 1 shows a particular nodes distribution, where both nodes 1 and 2 (marked as the solid circle) are on the boundary and three interior nodes (marked as the open circle) are used to calculate the partial derivatives with respect to y on nodes 1 and 2, respectively. Approximation of the derivatives $\partial u/\partial y$ from node distribution of node 2 is expected to be more accurate and stable than the result from node 1. This is due to the fact that the three interior nodes above node 2 are strictly lined up in the direction of y axis. The same idea can be applied to calculate second derivatives $\partial^2 u/\partial y^2$ or $\partial^2 u/\partial x^2$. Since the typical partial differential equation is governed by partial derivatives of x and y, it is ideal to have node distributed along the direction parallel to x and y axis. If the governing equation is formed as derivatives of x or y with various orders, nodes distributed on the line that parallel to the axis x or y will lead to a better result. Similarly, if the governing equation was formed with by spherical coordinates, then nodes should be more uniformly distributed along the sphere or the angle.

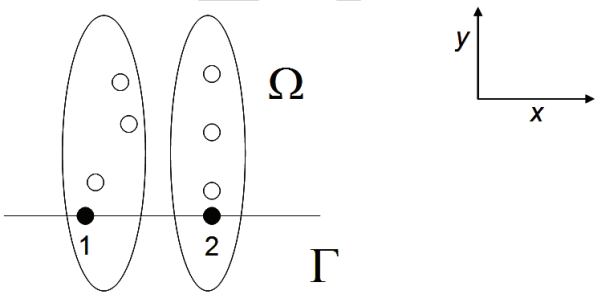

Figure 1: Left: The boundary point (node 1, which is marked as solid circle) and the interior points (marked as open circles) are distributed arbitrary. Right: The boundary point (node 2, which is marked as solid circle) and interior points (marked as open circles) are aligned in the direction of y-axis.

Gray Proof

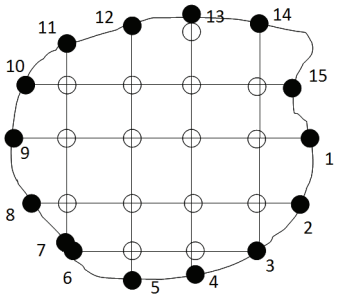

Figure 2: Uniformly distributed interior nodes (white ○) and boundary nodes (solid •) generated by the CKD.

In the numerical implementation, for a better accuracy it is important to distribute nodes uniformly so that they are parallel to the x and y axes, as shown in Fig. 2. This is called the cross knot distribution (CKD). The general guide of CKD is to try to generate a uniform mesh grids to cover the domain and then find the intersection nodes of mesh grids and boundary. The solid black nodes of #1-#15 on the boundary are these intersection nodes. As we have noticed that there is a strong relation in terms of the location among all interior and boundary nodes. For example, the pair of nodes (#1,#9) and (#5,#12) are lined up on the opposite sides of the boundary in the horizontal and vertical directions, respectively. This is a sharp contrast to the traditional approach that the boundary and interior nodes are generated independently. In such cases, it is not clear how to choose the numbers of boundary and interior nodes.

Numerical experiments shown in the next section indicate that when the interior and boundary nodes are distributed uniformly as shown in Fig. 2, the approximated partial derivatives are more stable than the traditional node distributions (uniformly node distributions on the boundary of the domain and inside of the domain, separately). The CKD can be easily built up by using the following four steps: 1. generate uniformly distributed grid nodes; 2. extract the inner nodes by check the boundaries of the domain; 3. extend the grid nodes inside the domain but near the boundary to generate the cross nodes between the extended axes and the boundary; 4. check if there are some overlap nodes. Note that in Fig. 2 there are two nodes #6 and #7 are very much close to each other, so the boundary nodes generated by CKD need to be pre-checked before their usages. When two nodes are very close, they might have bad influence on the numerical results. Only one node is employed to avoid the situation like #6 and #7. The CKD can normally enhance the stability and accuracy of the problem.

Next, we consider how to choose a good shape parameter of RBFs in the LMAPS. Although the process can be applied to all methods that involve RBF collocations. First, let us consider a companion differential equation of Eqs. (2.1a)-(2.1b) with a known exact solution $v(x,y)$; i.e.,

$$Lv(x,y)=\tilde{f}(x,y), \qquad (x,y)\in\Omega, \tag{3.1a}$$

Galley Proof

$$Bv(x,y)=\tilde{g}(x,y), \qquad (x,y)\in\partial\Omega. \tag{3.1b}$$

There are various ways to find a good shape parameter for the RBF methods for such equations with known exact solutions. The goal is to select a shape parameter which leads not only better accuracy in the RBF collocation but also "smooth" in the error distribution. After that, the same selected shape parameter is used to solve Eqs. (2.1a)-(2.1b).

4 Numerical results

To show the effectiveness of the proposed methods, we give ~~two~~ examples in 2D. For the implementation of the LMAPS, each influence domain contains 9 nearest neighboring points. MQ is selected as the basis function. The numerical computations in this section were carried out using MATLAB© on a desktop PC with 8x Intel(R) Core(TM) i7-2600k CPU@3.40 GHZ, 16 GB memory, in Linux OS Ubuntu 14.04.1 LTS.

Example 4.1. We first consider Poisson equation with the Dirichlet boundary condition as follows

$$\Delta u(x,y)=f(x,y), \qquad (x,y)\in\Omega, \tag{4.1a}$$
$$u(x,y)=g(x,y), \qquad (x,y)\in\partial\Omega, \tag{4.1b}$$

where Ω is the inner domain, $\partial\Omega$ is the whole boundary and g is given as the exact solution $u(x,y)=\exp(x-y)$ and f is the forcing term of the governing equation which can be derived from the exact solution.

The profile of computational domain is a gear-shaped domain as shown in Fig. 3, which is given by the following parametric equation:

$$\partial\Omega=\{(x,y)\,|\,x=r(\theta)\cdot\cos(\sigma(\theta)),\ y=r(\theta)\cdot\sin(\sigma(\theta)),\ 0\le\theta<2\pi\}, \tag{4.2a}$$
$$r(\theta)=2+\frac{1}{2}\sin(8\theta)), \qquad \sigma(\theta)=\theta+\frac{1}{5}\sin(8\theta). \tag{4.2b}$$

To validate the contribution of the proposed node arrangement CKD, we solve Eq. (4.1a) with two different types of nodes distribution. The left graph in Fig. 4 shows the uniformly distributed nodes with 700 boundary nodes and 1980 interior nodes. On the left of Fig. 5, there are 382 boundary nodes which are distributed by using the proposed CKD with the same 1980 interior points. The right of Fig. 4 and 5 show the performance of the LMAPS, in which the nodes distribution by CKD is greatly improved compared to the uniform node distribution.

To find a good shape parameter, let us assume that the exact solution of the above differential equation is not available. Consider a test differential equation with the same differential operator as Eq. (4.1a),

$$\Delta u(x,y)=0, \qquad (x,y)\in\Omega, \tag{4.3a}$$
$$u(x,y)=x+y, \qquad (x,y)\in\partial\Omega, \tag{4.3b}$$

Galley Proof

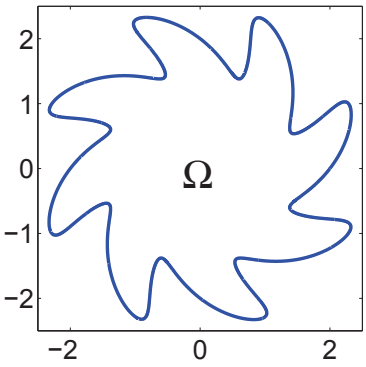

Figure 3: The profile of the computational domain.

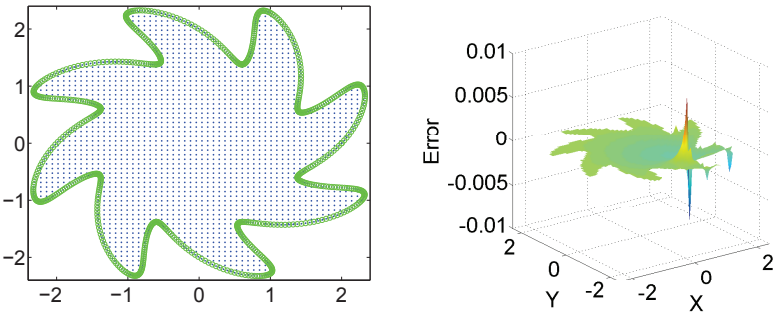

Figure 4: Uniform nodes distribution (left) and the absolute errors (right) using $c=2$ and 9 local nodes.

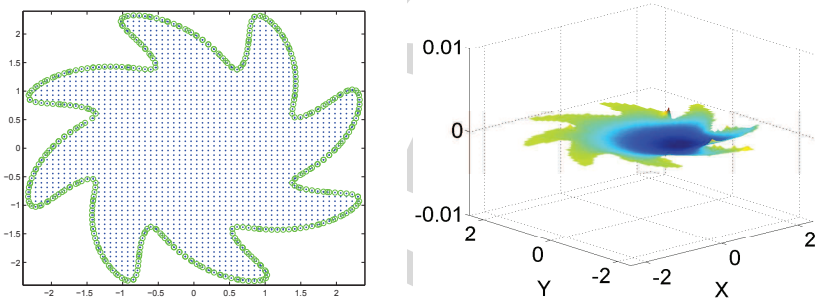

Figure 5: Nodes distribution generated by CKD (left) and the absolute errors (right) using $c=2$ and 9 local nodes.

in which the exact solution is known $u(x,y)=x+y$. The goal is to find a suitable shape parameter for the MQ RBFs so that the error of the approximated solution of the test differential equation is small and smooth. Subsequently, the same RBF shape parameter is being used to solve Eq. (4.1a).

Galley Proof

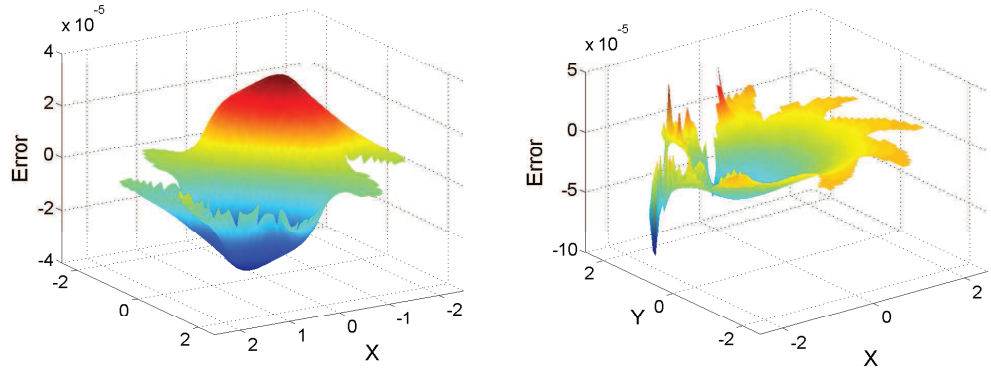

Figure 6: The profiles of the absolute errors for solving Eq. (4.3a) on the left and Eq. (4.1a) on the right using $c=1$.

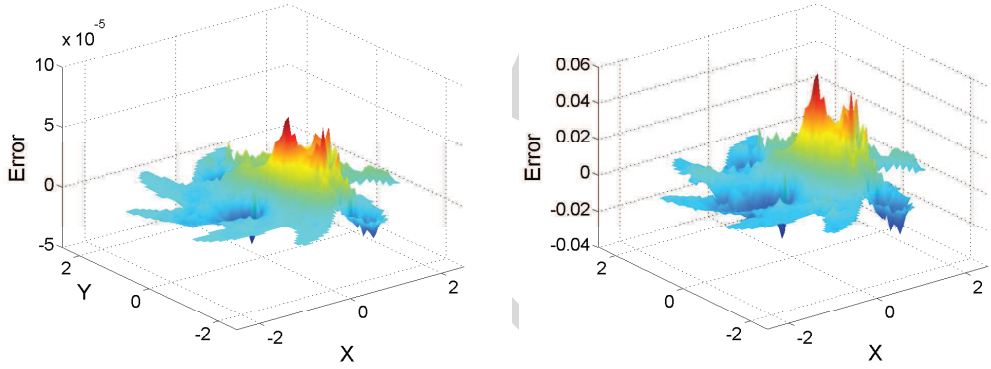

Figure 7: The profiles of the absolute errors for solving Eq. (4.3a) on the left and for Eq. (4.1a) on the right using $c=5$.

To test the effect of shape parameters on the performance of the LMAPS, we choose 1966 uniformly distributed interior nodes and 382 boundary nodes with various shape parameters. Fig. 6 shows the distributions of the absolute errors with $c=1.0$ for Eq. (4.3a) on the left and for Eq. (4.1a) on the right, respectively. We observe that the error distribution of the test differential equation in Fig. 6 is relatively smooth and the error of the original differential equation is acceptable. We obtain the similar results for the shape parameter ranging from 0.5 to 2.0.

For $2<c<10$, the error distribution of the test differential equation is non-smooth as shown in Fig. 7 on the left. Using the same shape parameter for solving the original equation, the obtained accuracy as shown in Fig. 7 on the right is significantly deteriorated. When the shape parameter is too large, there are sharp spikes in the error plot as shown in Fig. 8 on the left which is obtained by the test differential equation with $c=10$. The accuracy of the original differential equation is not acceptable, as shown in Fig. 8 on the right, using the same shape parameter as the test differential equation.

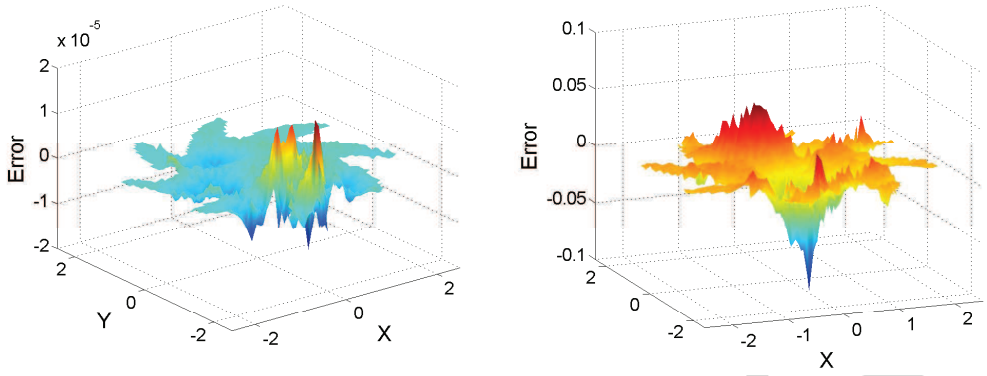

Figure 8: The profiles of the absolute errors for solving Eq. (4.3a) on the left and for Eq. (4.1a) on the right using $c=10$.

By comparing Figs. 6-8, we observe that if the profiles of the absolute errors of the test differential equation are smooth, then the results of using the same shape parameter for the original problem are stable (the profile of the absolute errors are less bumpy) and relatively accurate (similar the accuracy for the test problems). On the other hand, if the erratic behavior of the profile of the absolute errors occurs when solving test differential equation, then the selected shape parameter may not be suitable for solving the original differential equation.

To show that the proposed method for selecting a good shape parameter by using a test differential equations can be combined with other existed methods for selecting shape parameters, the Leave-One-Out Cross Validation (LOOCV) algorithm is performed and tested in this paper. We refer readers to the following [4,14] for the further details for LOOCV. In the implementation of LOOCV, an initial search interval [min,max] is needed. If the bounds selected are not reasonable, the algorithm LOOCV can perform incorrectly and produce a wrong estimation to the optimal shape parameter. In Table 1, we choose the lower bound of search interval $\min=0$ and we observe that the results using $\max=$ 3,4,5,6 and 10 are consistent and good, while unacceptable results are obtained using

Table 1: Find a suitable shape parameter c using LOOCV with various search intervals.

max	c	Maximum Error
3	1.8144	1.016E-04
4	1.8137	9.983E-05
5	1.8151	1.071E-04
6	1.8132	1.037E-04
7	6.0574	1.956E-02
8	6.0559	1.355E-02
9	6.0639	4.728E-02
10	1.8148	8.978E-05

max = 7,8 and 9. It is clear that the upper bound max can be hard to choose, especially in this case, where [0,10] performs the best, even though [0,7], [0,8] and [0,9] are not. Thus, when LOOCV is preferred but a suitable bound is not clear, the "good" shape parameters for the test differential equations can be used to narrow the search range for LOOCV.

Example 4.2. Second, we will consider the same Poisson equation as shown in Example 4.1 with mixed boundary conditions as follows

$$\Delta u(x,y) = f(x,y), \qquad (x,y) \in \Omega, \tag{4.4a}$$
$$u(x,y) = g(x,y), \qquad (x,y) \in \partial\Omega|_{x\geq 0}, \tag{4.4b}$$
$$\frac{\partial u(x,y)}{\partial \mathbf{n}} = h(x,y), \qquad (x,y) \in \partial\Omega|_{x<0}, \tag{4.4c}$$

where $\partial\Omega$ is separated as two parts $x<0$ and $x\geq 0$, $\partial\Omega|_{x\geq 0}$ satisfies the Dirichlet boundary condition and $\partial\Omega|_{x<0}$ satisfies the Neumann boundary conditions, $\mathbf{n}$ is the normal vector, h is the normal derivative of the exact solution $u(x,y) = \exp(x-y)$. Note that the computational domain is the same as shown in Example 4.1 as well. The test differential equation for the Poisson's equation with the mixed boundary conditions is

$$\Delta u(x,y) = 0, \qquad (x,y) \in \Omega, \tag{4.5a}$$
$$u(x,y) = x+y, \qquad (x,y) \in \partial\Omega|_{x\geq 0}, \tag{4.5b}$$
$$\frac{\partial u(x,y)}{\partial \mathbf{n}} = n_x + n_y, \qquad (x,y) \in \partial\Omega|_{x<0}, \tag{4.5c}$$

where the exact solution is given by $u(x,y) = x+y$ for $(x,y) \in \Omega \cup \partial\Omega$, (n_x, n_y) is the normal vector.

To test the effect of shape parameters on the performance of the LMAPS, we choose 1966 uniformly distributed interior nodes and 382 boundary nodes with various shape parameters. Fig. 9 shows the distributions of the absolute errors with $c=2$ for Eq. (4.5) on the left and for Eq. (4.4) on the right, respectively. Fig. 10 and Fig. 11 show the similar results with $c=4$ and with $c=5$, respectively. Similar observation can be obtained compared to Example 4.1. That says, when a shape parameter produce a smooth and relatively accurate results for the test differential equations using LMAPS, the same shape parameter will not perform very inaccurately on the original differential equations. However, when the error profile from the test problem is already not smooth, even if it is accurate (Fig. 10), the same shape parameter will not perform well on the original problem. Additionally, if a shape parameter does not perform well on the test problem, there is little or no hope that shape parameter can perform well on the original problem (Fig. 11).

To test performance of LOOCV, we select several bounds as well in this example. In Table 2, we choose the lower bound of search interval min = 0 and vary the upper bound. We observe that the results using max = 3,4 and 5 are consistent and good while unacceptable results are obtained using max = 6,7,8,9 and 10. Using the proposed approach to initially identify the range of good shape parameter, we can then apply LOOCV to provide a good initial search interval to further identify the optimal shape parameter.

Galley Proof

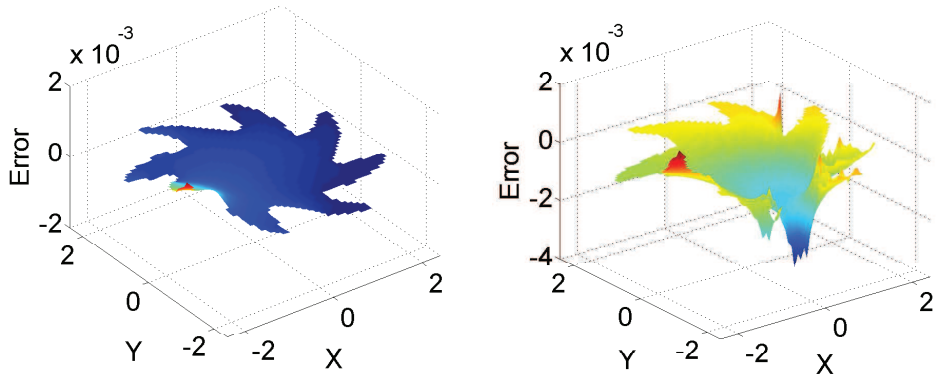

Figure 9: The profiles of the absolute errors for solving Eq. (4.5) on the left and for Eq. (4.4) on the right using $c=2$.

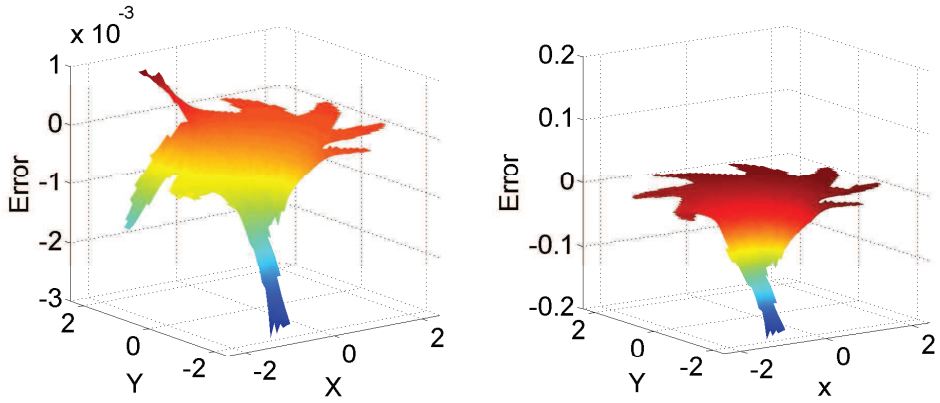

Figure 10: The profiles of the absolute errors for solving Eq. (4.5) on the left and for Eq. (4.4) on the right using $c=4$.

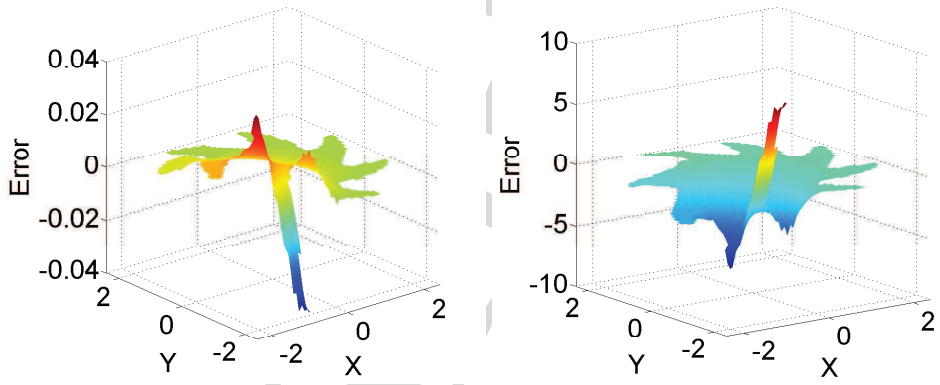

Figure 11: The profiles of the absolute errors for solving Eq. (4.5) on the left and for Eq. (4.4) on the right using $c=5$.

Table 2: Find a suitable shape parameter c using LOOCV with various search intervals.

max	c	Maximum Error
3	2.6730	1.353E-02
4	2.6745	3.834E-03
5	2.6755	7.058E-04
6	5.9962	3.156E-00
7	6.9950	2.676e-00
8	7.9941	4.723E-00
9	8.9934	1.046E-00
10	9.9940	1.179E-00

Example 4.3. We further test the proposed methods on the following eigenvalue problem

$$\Delta u(x,y) = -\lambda u, \qquad (x,y) \in \Omega, \tag{4.6a}$$

$$u(x,y) = 0, \qquad (x,y) \in \partial\Omega, \tag{4.6b}$$

where $\Omega = [0,1] \times [0,1]$. The analytic eigenvalues and eigenvectors are given by

$$\lambda_{ij} = (i^2 + j^2)\pi^2, \quad u_{ij}(x,y) = \sin(i\pi x)\sin(j\pi y), \quad i,j = 1,2,3,\cdots. \tag{4.7}$$

To find the eigenvalues numerically using LMAPS, rewrite the eigenvalue problem by $-\Delta u = \lambda u$, where the coefficients matrix in the sparse system Eqs. (2.11a)-(2.11b) is generated by following the same procedure for solving Laplace equation of the form $-\Delta u = 0$. The MATLAB built-in function `eigs` is then used to find the eigenvalues of the sparse coefficient matrix. Similar to Example 4.1, we choose the test differential equation (4.3a) on the unit square with $g(x,y) = x$. The exact solution is given as $u(x,y) = x$. For the numerical implementation, 20×20 nodes in the unit square were selected. For $c = 1$, the error profile for solving the test differential equation is given by Fig. 12. Using the same shape parameter, the first eight eigenvalues of Eq. (4.6a) are shown in Table 3. We have observed the similar connection between the smoothness of the error profile in Fig. 12 for the test problem on the left and the accurate approximation of eigenvalues in Table 3 as shown in Example 4.1.

On the other hand, we choose a much larger shape parameter $c = 10$ in our implementation for the test differential equation Eq. (4.3a) and the sought differential equation Eq. (4.6a). As shown in Fig. 12 on the right, despite a better accuracy than the case using $c = 1$ for the test differential equation, the accuracy of the eigenvalue problem becomes

Table 3: The first eight eigenvalues of Eq. (4.6a) using $c = 1$.

n	1	2	3	4	5	6	7	8
Exact	19.74	49.35	49.35	78.96	98.70	98.70	128.30	128.30
Numerical	19.74	49.34	49.34	78.97	98.57	98.58	128.27	128.28

Galley Proof

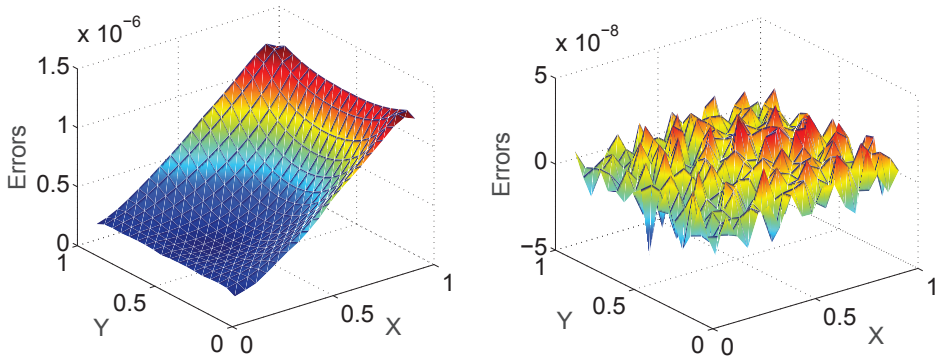

Figure 12: Example 4.2: The error profiles of Eq. (4.3a) in the unit square with $g(x,y)=x$ using $c=1$ on the left and $c=10$ on the right.

Table 4: The first eight eigenvalues of Eq. (4.6a) using $c=10$.

n	1	2	3	4	5	6	7	8
Exact	19.74	49.35	49.35	78.96	98.70	98.70	128.30	128.30
Numerical	7.52	19.31	19.64	25.66	25.66	29.65	29.65	31.44

worse as shown in Table 4. As we have seen in Fig. 12 on the right, many sharp spikes have appeared in the error profile. This is an indication that the smoothness of the obtained results for the test differential equation has an impact on stable solution of the given eigenvalue problem. By comparing Table 3 with Table 4, we find that if the test errors are smoothly distributed over the domain, then the results for the original given problem are expected to be stable.

5 Conclusions

In this paper, we propose a new approach for the selecting node distribution and how to choose a good shape parameter of RBFs for solving partial differential equations using the LMAPS and the concept of test differential equations.

The node distribution suggested in this paper is similar to the typical finite difference techniques, where nodes on the boundary is suggested to be aligned with the nodes inside the computational domain in each axis. This is what we called *cross knot distribution (CKD)*.

The test problem associated with the given differential equations are defined as an equation with the same differential operator and the same boundary conditions as the given equation, but with an *simple* known solution. In the numerical experiments we show some evidence of the smoothness of the error distribution for the test problem provides a way to choose a good shape parameter of RBFs for the original problem.

This paper tested only several differential equations with Laplacian differential op-

Galley Proof

erator in 2D, Δ. Thus, the simplest known solution to a Laplace equation with whether Dirichlet or Neumann boundary conditions, or even mixed boundary conditions, would be solutions of the form $u(x,y)=ax+by$. Due the linearly property of the equations involved, these simple known solutions will behavior similar to each other. Thus, we tested test problem with solution $u(x,y)=x+y$ in Example 4.1 and Example 4.2 and $u(x,y)=x$ in Example 4.3. We noticed that the behavior of the shape parameter in test problem (smoothness and accuracy) effected the behavior of the shape parameter in the original differential equations, one reason may be due to the similar behavior of the solutions to the test problem and to the original problem.

The readers may noticed in [11], authors employed a node scalling technique based on the largest distance in each axis direction in the local domains, so the best shape parameter for multiquadric RBF in their experiments is suggested to be approximately 30. The techniques in this paper on selecting a good shape parameter is based on test differential equations on a complicated domain as shown in the original domain, without scalling parameter. Both scalling and domain shape can effect the choice of the "best" shape parameter can be used in the problems. This needs our further investigations.

Additionally, for problems in three dimensional space it is a challenge to specifically locate the boundary nodes so that they are correlated with the interior nodes. Further applications of the proposed method to solving real engineering problems are subject to future investigation.

Acknowledgements

Financial supported from the National Natural Science Foundation of China (No. 1702125) and the China Scholarship Council are gratefully acknowledged.

References

[1] RALPH E. CARLSON AND THOMAS A. FOLEY, *The parameter {R2} in multiquadric interpolation*, Comput. Math. Appl., 21(9) (1991), pp. 29–42.

[2] C. S. CHEN, C. M FAN AND P. H. WEN, *The method of particular solutions for solving elliptic problems with variable coefficients*, Int. J. Numer. Methods Biomedical Eng., 8 (2011), pp. 545–559.

[3] C. S. CHEN, C. M. FAN AND P. H. WEN, *The method of particular solutions for solving certain partial differential equations*, Numer. Methods Partial Differential Equations, 28 (2012), pp. 506–522.

[4] G. E. FASSHAUER AND J. G. ZHANG, *On choosing optimal shape parameters for RBF approximation*, Numer. Algorithms, 45(1-4) (2007), pp. 345–368.

[5] THOMAS A. FOLEY, *Interpolation and approximation of 3-d and 4-d scattered data*, Comput. Math. Appl., 13(8) (1987), pp. 711–740.

[6] R. FRANKE, *Scattered data interpolation: tests of some methods*, Math. Comput., 38 (1982), pp. 181–200.

Geujroof

[7] M. A. GOLBERG, C. S. CHEN AND S. KARUR, *Improved multiquadric approximation for partial differential equations*, Eng. Analy. Boundary Elements., 18(1) (1996), pp. 9–17.
[8] R. L. HARDY, *Multiquadric equations of Topography and other irregular surfaces*, J. Geophys. Res., 176 (1971), pp. 1905–1915.
[9] E. J. KANSA, *Multiquadrics-a scattered data approximation scheme with applications to computational fluid dynamics-II*, Comput. Math. Appl., 19(8/9) (1990), pp. 147–161.
[10] E. J. KANSA AND Y. C. HON, *Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations*, Comput. Math. Appl., 39(7-8) (2000), pp. 123–137.
[11] G. KOSEC AND B. SARLER, *Local rbf collocation method for darcy flow*, CMES: Comput. Model. Eng. Sci., 25(3) (2008), pp. 197–208.
[12] E. LARSSON AND B. FORNBERG, *A numerical study of some radial basis function based solution methods for elliptic PDEs*, Comput. Math. Appl., 46 (2003), pp. 891–902.
[13] C. K. LEE, X. LIU AND S. C. FAN, *Local multiquadric approximation for solving boundary value problems*, Comput. Mech., 30 (2003), pp. 396–409.
[14] S. RIPPA, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math., 11 (1991), pp. 193–210.
[15] C. SHU, H. DING AND K. S. YEO, *Local radial basis function-based differential quadrature method and its application to solve two dimensional incompressible navier-stokeequations*, Comput. Methods Appl. Mech. Eng., 192 (2003), pp. 941–954.
[16] CHIA-CHENG TSAI, Z.-H LIN AND T.-W HSU, *Using a local radial basis function collocation method to approximate radiation boundary conditions*, Ocean Eng., 105(07) (2015), pp. 231–241.
[17] G. YAO, J. KOLIBAL AND C. S. CHEN, *A localized approach for the method of approximate particular solutions*, Comput. Math. Appl., 61(9) (2011), pp. 2376–2387.

Galley Proof