Computers and Mathematics with Applications 70 (2015) 2883–2894

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

The localized method of approximated particular solutions for near-singular two- and three-dimensional problems

Guangming Yao [a], C.S. Chen [b,c], Wen Li [b,a,*], D.L. Young [d]

[a] Department of Mathematics, Clarkson University, Potsdam, NY 13699, USA
[b] School of Mathematics, Taiyuan University of Technology, 030024, China
[c] Department of Mathematics, University of Southern Mississippi, Hattiesburg, MS 39406, USA
[d] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 17 April 2015
Received in revised form 14 September 2015
Accepted 26 September 2015
Available online 29 October 2015

Keywords:
Method of approximate particular solutions
Meshless method
Near-singular problem
MQ radial basis functions
Shape parameter
Kansa's method

ABSTRACT

In this paper, the localized method of approximate particular solutions (LMAPS) using radial basis functions (RBFs) has been simplified and applied to near-singular elliptic problems in two- and three-dimensional spaces. The leave-one-out cross validation (LOOCV) is used in LMAPS to search for a good shape parameter of multiquadric RBF. The main advantage of the method is that a small number of neighboring nodes can be chosen for each influence domain in the discretization to achieve high accuracy. This is especially efficient for three-dimension problems. There is no need to apply adaptivity on node distribution near the region containing spikes of the forcing terms. To examine the performance and limitations of the method, we deliberately push the spike of the forcing term to be extremely large and still obtain excellent results. LMAPS is far superior than the compactly supported RBF (Chen et al. 2003) for such elliptic boundary value problems.

1. Introduction

During the past two decades, radial basis functions (RBFs) have became an effective tool for not only surface interpolation and data reconstruction but also for numerical partial differential equations (PDEs). The RBF collocation method was first introduced by Kansa in 1990 [1]. Since then, RBFs have been widely used for solving various kinds of science and engineering problems. Motivated by Kansa's method, several modified versions of RBF collocation methods [2–5] have been developed. Despite the effectiveness and simplicity of these methods, the resultant system of equations is often ill-conditioned, especially when a large amount of collocation points is needed. The selection of the optimal shape parameter of RBFs is also an outstanding research topic. The fundamental issue of the above problems is that the RBFs being used in these methods are globally supported functions. The global RBFs are effective when the solution space of given PDEs is reasonably smooth. On the contrary, when the solution space fluctuates rapidly or presents sharp spikes, the global methods generally do not work well.

Traditional methods such as the finite element method and finite difference method are local methods. They are very effective in dealing with the problems mentioned above. The idea of local collocation methods in the context of RBFs has been introduced by Lee et al. [6] and Tolstykh and Shirobokov [7]. Instead of using all of the interpolation points as in the global case, only data that falls in the influence domain of a given node is used for discretization of the PDE at the node. Based

* Corresponding author at: School of Mathematics, Taiyuan University of Technology, 030024, China.
E-mail address: muziwen02@163.com (W. Li).

http://dx.doi.org/10.1016/j.camwa.2015.09.028
0898-1221/

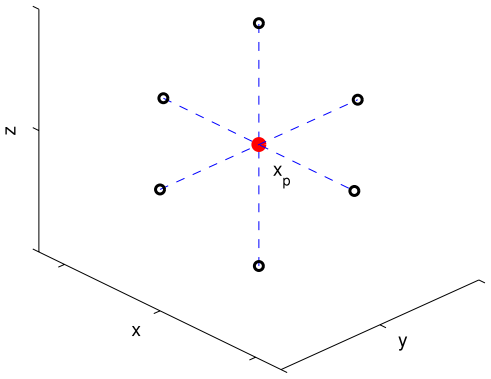

Fig. 1. The 7-point stencil of $\mathbf{x}_p$ in three-dimensional space.

on a similar idea, the global version of the method of approximate particular solutions (MAPS) [2,5] has been extended to the localized version (LMAPS) [8]. LMAPS is very effective for solving large-scale problems. As shown in [8], nearly one million nodes have been used for solving Poisson problems with excellent results. As expected, the selection of a good shape parameter of MQ in LMAPS is much easier than the selection in the global MAPS.

In this paper, we will make improvements to the previous version of LMAPS, and apply it to various near-singular problems in two- or three-dimensional space. A simplified formulation of the original LMAPS is introduced, where the boundary conditions are merged into the governing equations in the discretization process. The leave one out cross validation (LOOCV) [9,10] will be used to select good shape parameter in MQ. Due to the unique feature of the local approach, LMAPS is capable to capture the rapid variation of the solution. Hence, LMAPS is especially attractive for solving near-singular problems as shown in [11]. However, [11] used the smoothing scheme. The enhance nodes near the large spike of the forcing term were needed. The difficulty of solving such near-singular problems becomes more pronounced when the spike of the forcing term gets larger. As indicated in [11], the global methods using RBFs fail to produce acceptable accuracy for the near-singular problems. Chen et al. [11] coupled the method of fundamental solutions with the compactly supported RBFs (CS-RBFs). With the improved version of LMAPS shown in this paper, we are able to solve such near-singular problems without the tedious adaptive effort mentioned in [11], especially we are able to solve three-dimensional problems on extremely irregular domains.

The paper is organized as follows. In Section 2, we briefly extend LMAPS to three-dimensional space. This is a trivial extension due to the radial nature of RBFs. In Section 3, we introduce the simplified formulation of LAMPS. This improved the efficiency of the method. The particular solution of the MQ RBF for Laplace operator in 2D and 3D are given. In Section 4, we also briefly explain LOOCV for selecting a good shape parameter in MQ RBFs. In Section 5, to demonstrate the effectiveness of LMAPS for solving near-singular problems, we tested our methods on the following PDEs: two Poisson's equations on unit cube or irregular domains in 2D and 3D; and a modified Helmholtz problem on Stanford Bunny domain in 3D. In Section 6, we make some comments and draw conclusions.

2. LMAPS

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

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

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

where $\Omega \subset \mathbb{R}^2$ or $\mathbb{R}^3$ is a bounded and closed two- or three-dimensional domain with a boundary $\partial\Omega$. We assume that the above boundary value problem has a unique solution for sufficiently smooth nonhomogeneous term f and boundary data g.

Let $\{\mathbf{x}_j\}_{j=1}^{n}$ be the interpolation points inside the domain Ω. In this paper, the interpolation points are distributed uniformly in the domain. For any point $\mathbf{x}_p \in \Omega$, we create an influence domain Ω_p, which contains a region formed by the n_s nearest neighboring interpolation points $\{\mathbf{x}_j\}_{j=1}^{n_s}$ to $\mathbf{x}_p$ including $\mathbf{x}_p$. We usually choose five nearest neighboring points for each node in 2D. For three-dimensional problems, we intend to select more neighboring points, about 7–25 points, than that in two-dimensional domain for higher dimensional space. Figs. 1 and 2 show the seven and nineteen nearest neighboring interpolation points of $\mathbf{x}_p$, respectively, in 3D.

By the method of particular solutions (MAPS) [2], $u(\mathbf{x}_p)$ can be approximated by a linear combination of n_s RBFs in the following form:

$$u(\mathbf{x}_p) \simeq \hat{u}(\mathbf{x}_p) = \sum_{j=1}^{n_s} a_j \Phi\left(\left\|\mathbf{x}_p - \mathbf{x}_j\right\|\right), \tag{3}$$

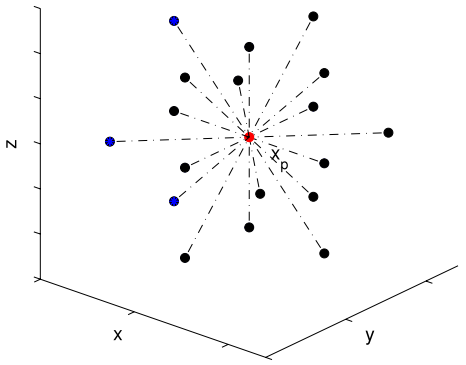

Fig. 2. The 19-point stencils of $\mathbf{x}_p$ in three-dimensional space.

where $\{a_j\}_{j=1}^{n_s}$ are coefficients to be determined and $\|\cdot\|$ is the Euclidean norm. Note that

$$L\Phi = \varphi \tag{4}$$

where φ is a RBF. Note that (3) holds for every point in Ω_p, specifically for these local points $\{\mathbf{x}_j\}_{j=1}^{n_s} \subset \Omega_p$. This implies the following system of equations

$$\hat{\mathbf{u}}_{n_s} = \mathbf{\Phi}_{n_s}\mathbf{a}_{n_s}, \tag{5}$$

where $\mathbf{\Phi}_{n_s} = \left[\Phi\left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|\right)\right]_{1\le i,j\le n_s}$, $\hat{\mathbf{u}}_{n_s} = [\hat{u}(\mathbf{x}_1), \ldots, \hat{u}(\mathbf{x}_{n_s})]^T$, $\mathbf{a}_{n_s} = [a_1, a_2, \ldots, a_{n_s}]^T$. It can be proved that the local matrix on the right-hand side of (5) is non-singular by using RBFs such as Gaussian, multiquadric and inverse multiquadric. Thus the inverse matrix can be computed provided that all the nodal interpolation points inside the influence domain Ω_p are distinct (we shall see later that we do not need to calculate inverses of the local matrices. Instead, we can solve a small linear system). The coefficient vector $\mathbf{a}_{n_s}$ in (5) can be expressed in terms of function values at those local interpolation points; i.e.,

$$\mathbf{a}_{n_s} = \mathbf{\Phi}_{n_s}^{-1}\hat{\mathbf{u}}_{n_s}. \tag{6}$$

In this way, instead of using the weighting coefficients $\mathbf{a}_{n_s}$, $\hat{u}(\mathbf{x}_p)$ in (3) can be expressed in terms of function values at n_s interpolation points, $\hat{\mathbf{u}}_{n_s}$; i.e.,

$$\begin{aligned}\hat{u}(\mathbf{x}_p) &= \sum_{j=1}^{n_s} a_j \Phi\left(\left\|\mathbf{x}_p - \mathbf{x}_j\right\|\right)\\ &= \left(\mathbf{\Theta}_{n_s}\mathbf{\Phi}_{n_s}^{-1}\right)\hat{\mathbf{u}}_{n_s}\end{aligned} \tag{7}$$

where

$$\mathbf{\Theta}_{n_s} = \left[\Phi\left(\left\|\mathbf{x}_p - \mathbf{x}_1\right\|\right), \ldots, \Phi\left(\left\|\mathbf{x}_p - \mathbf{x}_{n_s}\right\|\right)\right]. \tag{8}$$

Note that the coefficients $\left(\mathbf{\Theta}_{n_s}\mathbf{\Phi}_{n_s}^{-1}\right)$ can be obtained by solving a small linear system $\mathbf{\Phi}_{n_s}\alpha_{n_s} = \mathbf{\Theta}_{n_s}^T$, where α_{n_s} are the coefficients for $\hat{\mathbf{u}}_{n_s}$. We will notice that it is important to make use of (7) instead of (3) for the formulation of the local method.

To solve the boundary value problem (1)–(2), let $\mathbf{x}_p \in \Omega_p \subset \Omega$ be any interior point. From (1), we have

$$\begin{aligned}L\hat{u}(\mathbf{x}_p) &= \sum_{j=1}^{n_s} a_j L\Phi\left(\left\|\mathbf{x}_p - \mathbf{x}_j\right\|\right)\\ &= \left(L\mathbf{\Theta}_{n_s}\mathbf{\Phi}_{n_s}^{-1}\right)\hat{\mathbf{u}}_{n_s}.\end{aligned} \tag{9}$$

From (1) and (9), we have

$$\left(L\mathbf{\Theta}_{n_s}\mathbf{\Phi}_{n_s}^{-1}\right)\hat{\mathbf{u}}_{n_s} = f(\mathbf{x}_p), \quad \mathbf{x}_p \in \Omega_p. \tag{10}$$

It is easy to reformulate (10) in the global form

$$\left(L\mathbf{\Theta}_n\mathbf{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n = f(\mathbf{x}_p), \quad \mathbf{x}_p \in \Omega, \tag{11}$$

where

$$\hat{\mathbf{u}}_n = [\hat{u}(\mathbf{x}_1), \hat{u}(\mathbf{x}_2), \ldots, \hat{u}(\mathbf{x}_n)]^T,$$

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 positions. For further details of mapping (10)–(11), we refer the readers to [8].

For $\mathbf{x}_p \in \partial\Omega$, the formulation is the same as above by replacing the differential operator L in (9) with boundary operator B. Hence, we have

$$\left(B\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n = g(\mathbf{x}_p), \quad \mathbf{x}_p \in \partial\Omega. \tag{12}$$

Let $\{\mathbf{x}_i\}_{i=1}^{n_i}$ be the interior points, $\{\mathbf{x}_i\}_{i=n_i+1}^{n_i+n_b}$ be the boundary points, and $n = n_i + n_b$. From (11) and (12), the numerical discretization of (1)–(2) becomes

$$\left(L\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n = f(\mathbf{x}_j), \quad j = 1, 2, \ldots, n_i, \tag{13}$$

$$\left(B\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right)\hat{\mathbf{u}}_n = g(\mathbf{x}_j), \quad j = n_i + 1, \ldots, n. \tag{14}$$

3. Particular solutions and simplified formulation

In this section, we present a new improvement version of the original LMAPS by merging the boundary conditions into the governing equations. For the case of the Dirichlet boundary condition, (14) becomes

$$\hat{u}(\mathbf{x}_j) = g(\mathbf{x}_j), \quad j = n_i + 1, \ldots, n. \tag{15}$$

Then (13) and (15) can be rewritten in the following block matrix form

$$\begin{bmatrix} \boldsymbol{\Psi}_i & \boldsymbol{\Psi}_b \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_i \\ \hat{\mathbf{u}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \tag{16}$$

where $\begin{bmatrix} \boldsymbol{\Psi}_i & \boldsymbol{\Psi}_b \end{bmatrix} = L\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}$, and $\mathbf{0}$ is the zero matrix of size of $n_b \times n_i$, $\mathbf{I}$ is the $n_b \times n_b$ identity matrix, $\hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}_i & \hat{\mathbf{u}}_b \end{bmatrix}^T$, $\hat{\mathbf{u}}_i = [\hat{u}(\mathbf{x}_1), \ldots, \hat{u}(\mathbf{x}_{n_i})]^T$, and $\hat{\mathbf{u}}_b = [\hat{u}(\mathbf{x}_{n_{i+1}}), \ldots, \hat{u}(\mathbf{x}_n)]^T$. (16) can be further simplified to the following form

$$\boldsymbol{\Psi}_i\hat{\mathbf{u}}_i = \mathbf{f} - \boldsymbol{\Psi}_b\mathbf{g}. \tag{17}$$

We obtain the approximated solution of (1)–(2) at all of the internal nodal points by solving the above sparse system of equations. Note that (17) is a more compact form than (16). Instead of solving $n \times n$ matrix system in (16), we only solve a smaller $n_i \times n_i$ matrix system. For the case of the Neumann boundary condition, a similar formulation can be obtained.

In this paper we select MQ, $\varphi = \sqrt{r^2 + c^2}$, as the basis function in our implementation. The corresponding particular solution of Φ in (4) for $L = \Delta$ in the 2D case is as follows [2,5]

$$\Phi(r) = \frac{1}{9}\left(4c^2 + r^2\right)\sqrt{r^2 + c^2} - \frac{c^3}{3}\ln\left(c + \sqrt{r^2 + c^2}\right), \tag{18}$$

and the corresponding particular solution, Φ, for $L = \Delta$ in the 3D is as follows [2,5]

$$\Phi(r) = \begin{cases} \dfrac{2r^2 + 5c^2}{24}\sqrt{r^2 + c^2} + \dfrac{c^4}{8r}\ln\left(r + \sqrt{r^2 + c^2}\right) - \dfrac{c^4\ln(c)}{8r}, & \text{if } r \neq 0, \\ \dfrac{c^3}{3}, & \text{if } r = 0, \end{cases} \tag{19}$$

where r is the Euclidean distance.

For the modified Helmholtz operator, $L = \Delta - \lambda^2$, the numerical discretization of (1)–(2) can be formalized as

$$\left(L\boldsymbol{\Theta}_n\boldsymbol{\Phi}_n^{-1}\right) = \boldsymbol{\varphi}_n\boldsymbol{\Phi}_n^{-1} - \lambda^2 I, \tag{20}$$

where

$$\boldsymbol{\varphi}_n = \left[\varphi\left(\|\mathbf{x}_p - \mathbf{x}_1\|\right), \ldots, \varphi\left(\|\mathbf{x}_p - \mathbf{x}_n\|\right)\right]. \tag{21}$$

Thus, the particular solutions for $L = \Delta$ remains valid. We refer the readers to [8] for more details.

4. LOOCV

The accuracy of the RBF-based methods greatly depends on the choice of the basis functions and the so-called shape parameter. Among them, multiquadrics (MQ) is one of the most popular and effective RBFs. In general, even with a small number of basis functions, MQ can still produce very accurate results. However, how to select a good shape parameter is still an outstanding research topic [12,10,13,14]. Despite these efforts, the selection for a suitable shape parameter for radial basis functions remains a challenge.

In [6,15], the authors found the choice of shape parameters of MQ in the local collocation method is less sensitive than the choices in the global methods. The optimal shape parameter depends on the density of nodes, the number of nodes in the collocation, and the function values at the nodes in the influence domain. In this paper, we will make improvements to LMAPS by introducing leave-one-out cross validation (LOOCV) [9,10] to select a good shape parameter of the MQ RBF.

LOOCV is an estimation of the generalized performance of a model trained on $n-1$ samples of data which has a high variance. The more collocation points we have, the better shape parameter c can be selected. For efficiency reasons, we apply LOOCV to the local small systems in (9). Since each equation in the global sparse linear system can be obtained by using a different RBF or a fixed RBF with a different shape parameter, we can choose to apply LOOCV to find the optimal shape parameter in some or all of the local systems. In the next section, we use a fixed RBF (MQ) with a fixed c for all local systems. It is certainly not cost-effective to apply LOOCV to each local system. For efficiency, the shape parameter c is determined by applying LOOCV to an arbitrary local system. We slightly modify the MATLAB code in [9] and apply it to our case. The MATLAB function `fminbnd` is used to find the minimum of a function of one variable within a fixed interval. Since `fminbnd` is a local minimizer, we need to provide an initial guess of the lower and upper bounds so that the search can be performed in the interval [`min`, `max`]. We refer the interested readers to [9] for further details. The major advantage of LOOCV is that we do not need the exact solution for the determination of a proper MQ shape parameter. Furthermore, there are fewer critical parameters which need to be selected in LOOCV.

5. Numerical results

LMAPS is especially attractive for solving near-singular problems as shown in [11] where the smooth scheme and adaptive nodes near the large spike of the forcing term are needed. The difficulty of solving such near-singular problems becomes more pronounced when the spike of the forcing term gets larger. As indicated in [11], the global methods using RBFs fail to produce an acceptable approximation for the near-singular problems. Chen et al. [11] coupled the method of fundamental solutions, with a local basis, CS-RBF, to overcome the difficulties. A common feature of CS-RBF and LMAPS is that they both are localized meshless methods. However, as we shall see in this section, LMAPS is far superior than the CS-RBF for solving near-singular problems.

To test the effectiveness of the proposed algorithm on PDEs with near-singular forcing terms, we selected two Poisson problems and a modified Helmholtz problem in 2D and 3D. In all the numerical examples, the local formulation is done on a five-point neighborhood of each node for 2D problems, and a fifteen-point to twenty-point neighborhood for 3D problems. All the numerical results are compared with the analytical solutions, and with the results obtained in [11,16]. Since the maximum absolute value of the analytical solution could be extremely large and the minimum value could be very small, we will use the root mean square error (RMSE), the relative error (RE), the maximum relative absolute error (MRAE), and/or the maximum absolute error (MAE) to measure the numerical accuracy. The definition of these errors are as follows:

$$RMSE = \sqrt{\frac{1}{n_i}\sum_{k=1}^{n_i}\left(\hat{u}_k - u_k\right)}, \qquad RE = \frac{\|\hat{\mathbf{u}}_{n_i} - \mathbf{u}_{n_i}\|_2}{\|\mathbf{u}_{n_i}\|_2},$$

$$MRAE = \max_{1\le k\le n_i}\left(\frac{\left|\hat{u}_k - u_k\right|}{u_k}\right), \qquad MAE = \max_{1\le k\le n_i}\left(\left|\hat{u}_k - u_k\right|\right),$$

where n_i is the number of interior points. $\hat{u}_k = \hat{u}(\mathbf{x}_k)$ is the approximate solution, $u_k = u(\mathbf{x}_k)$ is the exact solution, $\mathbf{u}_{n_i} = \{u_1, u_2, \ldots, u_{n_i}\}$ and $\hat{\mathbf{u}}_{n_i} = \{\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{n_i}\}$. The interior interpolation points are generated by a uniform grid which can be produced by the MATLAB code meshgrid$(0:1/m:1)$ where m is the number of subdivisions in each axis direction. We recall n_i the number of interior points, n_b the number of boundary points, and $n = n_i + n_b$. In order to search the local neighboring points more efficiently, we choose the kd-tree algorithm which is widely available on the Internet. LOOCV is employed to automatically select a good shape parameter. The initial search interval using `fminbnd` should be chosen with care.

The computation is performed using MATLAB on a notebook computer with Intel Core i5 in Windows 7 Home Basic 64 bit.

Example 1. Consider the following near-singular Poisson problem in a unit cube (we extend the example in [11] from 2D to 3D),

$$\Delta u(x,y,z) = f(x,y,z), \qquad (x,y,z) \in \Omega, \tag{22}$$

$$u(x,y,z) = g(x,y,z), \qquad (x,y,z) \in \partial\Omega, \tag{23}$$

Table 1
The errors in Example 1 for $a = 5.197$, $b = -3.164$, $n_s = 25$.

n	RMSE	RE	MRAE	c
1,331	3.48E−2	2.30E−3	1.71E−2	2.994
4,096	2.69E−2	1.78E−3	5.93E−2	2.335
9,261	3.86E−3	2.55E−4	3.14E−2	1.721
17,576	4.25E−3	2.81E−3	9.23E−2	1.493

Table 2
The RMSE, RE and MRAE in Example 1 with $n_i = 6859$, $n_b = 2402$, $n_s = 25$, and $c = 1.721$ for various values of a with $b = -3.364$ and various b with $a = 5.197$.

a	RMSE	RE	MRAE	max $\|f\|$
5.197	1.87E−2	9.46E−4	2.73E−2	149,320
5.207	1.73E−2	8.74E−4	7.81E−3	29,744
5.217	1.73E−2	8.73E−4	1.70E−2	29,824
5.227	1.74E−2	8.71E−4	8.37E−3	29,905
b	RMSE	RE	MRAE	max $\|f\|$
−3.364	1.87E−2	9.46E−4	2.73E−2	149,320
−3.264	5.15E−3	3.01E−4	2.74E−2	141,605
−3.164	3.86E−3	2.55E−4	3.14E−2	134,649
−3.064	3.64E−3	2.67E−4	3.52E−2	128,344

where the domain is a unit cube $\Omega \cup \partial\Omega = [2, 3]^3$. The domain is shifted from the original to avoid small analytical solutions. The forcing term

$$f(x, y) = \frac{15r^4 + r^3(42b - 24a) + r^2(8a^2 - 68ab + 35b^2) + rab(24a - 60b) + 24a^2b^2}{4(r + b)^3(a - r)^{3/2}} - 500(x - y)^2$$

and

$$g(x, y) = \frac{r^2\sqrt{a - r}}{r + b} - \frac{500(x - y)^4}{24},$$

where $r = \sqrt{x^2 + y^2 + z^2}$, and a and b are constants. The exact solution is given by

$$u(x, y) = \frac{r^2\sqrt{a - r}}{r + b} - \frac{500(x - y)^4}{24}, \quad (x, y, z) \in \Omega \cup \partial\Omega.$$

Clearly, the singular points for $f(x, y, z)$ are $r = a$ and $r = -b$, which are close to the corner of the boundary points $(2, 2, 2)$ and $(3, 3, 3)$. Therefore, there are two sharp spikes near the points at $f(2, 2, 2)$ and $f(3, 3, 3)$, where $r = \sqrt{12} \approx 3.4641$ and $\sqrt{27} \approx 5.1962$, respectively.

In [11], due to the difficulty caused by the sharp spike around $f(3, 3, 3)$ and despite the fact that a local method, compactly supported RBF, was used, a special smoothing scheme with enhanced points (adaptive grid) still had to be implemented to obtain an acceptable approximate solution. After using all the tricks, the results obtained in [11] only reach an order of accuracy of $E - 3$. Using LMAPS, we can easily apply a large number of interpolation points to solve the problem with high efficiency and accuracy. For 2D case in [11], we can reach an order of accuracy of $E - 5$ (data not shown). Some test results in 3D are shown in Table 1. The shape parameter of MQ is obtained using LOOCV with initial search interval $[0, 3]$. With an increasing number of interpolation points the condition of the global sparse system did not get worse. Therefore, our numerical approximation remains accurate. For $n = 17{,}576$, it takes only 2.56 s of elapsed time to run the test. The accuracy remains high even when more interpolation points used.

Table 2 shows the errors and the maximum absolute values of the forcing term with respect to various a and b. In the localization process, 25 nearest neighboring points are used. We see that the numerical errors for the case of $a = 5.197$ are very much close to the ones from the case of $a = 5.227$.

Table 3 shows the optimal shape parameter obtained on different search interval [min, max] and corresponding errors when $n_i = 6859$, $n_b = 2402$, $n_s = 25$ for $a = 5.197$ and $b = -3.164$. The errors are very stable for different choice of search interval in terms of searching criteria of LOOCV. Fig. 3 shows the sensitivity of the relative error on the shape parameter c in MQ. When $c \in [0.5, 3]$, the relative errors are reasonably close to the optimal results suggested by LOOCV on search range $[0, 3]$. In other words, the performance of our method is not very sensitive to the selection of the search range and shape parameter. However, without LOOCV, we will still need multiple simulations to find the 'best' results. Overall, our approach is not only simple, accurate, but also very efficient.

Table 3
The RMSE, RE and MRAE in Example 1 using LOOCV on different search range [min, max] with $n_i = 6859$, $n_b = 2402$, $n_s = 25$ for $a = 5.197$ and $b = -3.164$.

[min, max]	RMSE	RE	MRAE	c
[0, 2]	3.30E−3	2.18E−4	8.12E−2	1.786
[0, 3]	3.86E−3	2.55E−4	3.14E−2	1.721
[0, 4]	4.26E−3	2.81E−4	3.70E−2	2.158

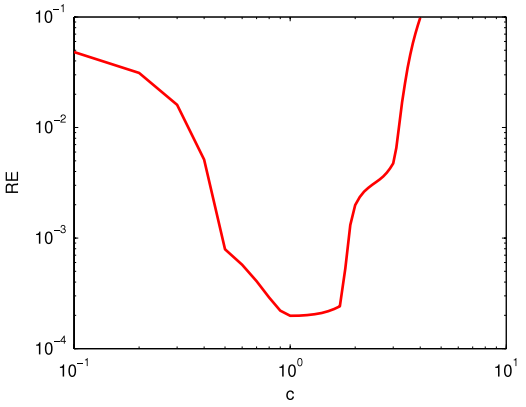

Fig. 3. The relative error RE versus shape parameter c in Example 1.

Example 2. We consider the same Poisson problem as shown in Example 1 with [11]

$$g(\mathbf{x}) = \frac{r^2}{r - a}, \tag{24}$$

in two- or three-dimensional space. We choose [0, 3] as the LOOCV initial search interval for all domains in this example.

- **Unit Square Domain in 2D**. We first consider a unit square domain, where $\mathbf{x} = (x, y)$. Let $r = \sqrt{x^2 + y^2}$. The exact solution is given by

$$u(x, y) = \frac{x^2 + y^2}{\sqrt{x^2 + y^2} - a}. \tag{25}$$

The forcing term is known

$$f(\mathbf{x}) = \frac{4a^2 - 3ar + r^2}{(r - a)^3}. \tag{26}$$

From (26) we observe that the forcing term f of (22) presents a singularity at the corner (1, 1) when $a = \sqrt{2}$. As a gets close to $\sqrt{2}$, the effect of the sharp spike near the corner (1, 1) is more pronounced. For instance for $a = 1.5$, the range of f is [3/8, 7343]. As a result, numerical approximation becomes more difficult. As shown in [11], the globally defined RBFs for solving this problem produces unacceptable results. For locally defined RBFs such as CS-RBFs, the accuracy is barely satisfactory [11].

To obtain accurate numerical approximations, a large number of interpolation points is usually suggested. We choose to use very dense interpolation points but keep the five point localization scheme. Since all interpolation points are uniformly distributed in the unit square domain, we only need to solve one local linear system of 5×5 for all the influence domains. Therefore, our algorithm is extremely efficient and the computational cost is not an issue at all even with a very large number of interpolation points. Using LOOCV, we get $c = 1.363$. Since the optimal shape parameter is only related to the nodes distribution, all the results in Table 4 are obtained with this optimal parameter with $n_i = 25,281$ and $n_b = 640$.

The errors increase when a approaches $\sqrt{2}$. This is due to the huge spike of forcing term when a gets closer to $\sqrt{2}$. In the table, max $|f|$ denotes the maximum absolute value of the forcing term f on grid points. It is amazing that despite the enormous spike of f, we still obtain very accurate results in order of magnitude of -3. Computation for each test case of a only takes about 2.0 s.

- **Irregular Domain in 2D**. To test the limitation of LMAPS on the near-singular problems, we consider the above problem with an irregular domain as shown in Fig. 4. The boundary $\partial\Omega$ is defined by the following parametric equation:

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

Table 4
The RMSE, RE and MRAE in Example 2 (in the unit square domain) with $n_i = 25,281$, $n_b = 640$ and $c = 1.363$ for various values of a.

a	RMSE	RE	MRAE	max $\|f\|$
1.43	5.64E−3	1.56E−3	3.28E−3	1,045,362
1.44	2.31E−3	7.09E−4	2.37E−3	244,074
1.45	1.27E−3	4.22E−4	1.94E−3	92,911
1.46	8.12E−4	2.89E−4	1.67E−3	45,133
1.47	5.69E−4	2.15E−4	1.50E−3	25,383
1.48	4.24E−4	1.69E−4	1.37E−3	15,744
1.49	3.29E−4	1.38E−4	1.27E−3	10,473
1.50	2.65E−4	1.16E−4	1.19E−3	7,343
1.60	6.69E−5	4.04E−5	8.27E−4	850

Table 5
The RMSE, RE and MRAE in Example 2 (in the 2D irregular domain) with $n_i = 27,346$, $n_b = 500$ and $c = 1.363$ for various values of a.

a	RMSE	RE	MRAE	max $\|f\|$
0.82	5.91E−2	1.48E−2	9.65E−2	640,414
0.81	1.67E−2	4.35E−3	4.48E−2	109,592
0.80	7.21E−3	1.92E−3	2.58E−2	35,650
0.78	2.23E−3	6.13E−4	1.16E−2	8,018
0.76	9.59E−4	2.70E−4	6.47E−3	2,868
0.74	4.90E−4	1.41E−4	4.06E−3	1,296
0.72	2.79E−4	8.20E−5	2.74E−3	675
0.70	1.72E−4	5.13E−5	1.95E−3	387

where

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

The singular point is close to the boundary in the lower left corner. The smallest r for all interpolation points is 0.83. Fig. 5 shows the profile of forcing term f for $a = 0.82$, and the range of the absolute values of f is [0.36, 640414]. We use 27,346 uniformly distributed points in the domain and 500 points on the boundary. A good shape parameter c selected by LOOCV is 1.363 From Table 5, we can see that the numerical results on such an irregular domain using LMAPS are still acceptable even for the near-singular case $a = 0.82$. Fig. 6 shows the profile of the relative absolute errors at given points with $a = 0.82$. The closer to the singular point, the larger the errors. Therefore, the analytical solution is stiff around the singular point.

Even though there are 27,346 small linear systems to be solved in this example, it only takes about 10 s to run a test case. There is no problem to increase the number of interpolation points to an even larger number. Consequently, we can efficiently solve large-scale problems with irregular domains by using LMAPS for similar kinds of problems.

- **Bumpy Sphere Domain in 3D**. Let Ω be a bumpy sphere as shown in Fig. 7. The functions f and g are given as follows

$$f(x, y, z) = \frac{2(3a^2 - 3ar + r^2)}{(r - a)^3},$$

where $r = \sqrt{x^2 + y^2 + z^2}$. The boundary $\partial\Omega$ of the bumpy sphere domain is defined as follows:

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

where

$$\rho = 1 + \frac{1}{6}\sin(6\theta)\sin(7\phi).$$

In Fig. 7, we can see the geometry of this 3D domain is quite complicated. It is not an easy task to solve problems in such a domain using traditional mesh-based methods. The pretreatment of finite element method and finite difference method is complicated for solving problems with this kind of domain. But for the meshfree method proposed in this paper, the process for solving this 3D problem is as simple as that for 2D problems with a square domain.

In the numerical computation, we choose 15 interpolation nodes in each influence domain in LMAPS. The value of forcing term f changes dramatically, from 0.77 to 256,680 for the case of $a = 0.46$. The results listed in Table 6 are obtained by using 28,120 interior nodes and 2500 boundary nodes. The profile of the absolute relative error on the surface of a bumpy sphere for $a = 0.40$ is shown in Fig. 8. As expected, errors are larger in the area near the singular point.

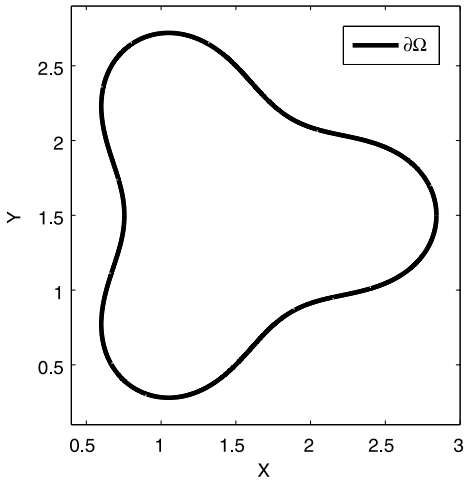

Fig. 4. The boundary of the 2D irregular domain in Example 2.

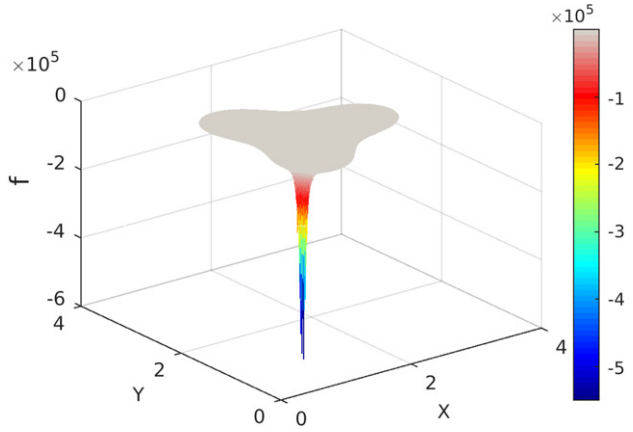

Fig. 5. Profile of the forcing term $f(x, y)$ for $a = 0.82$ in Example 2.

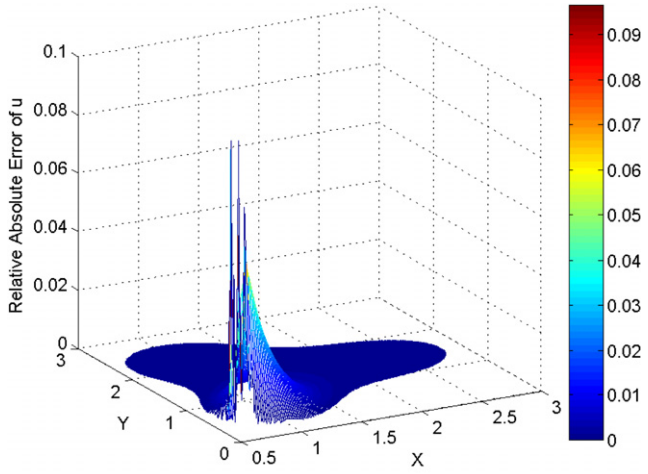

Fig. 6. The relative absolute errors at given points in Example 2 with $a = 0.82$ on the 2D irregular domain.

Example 3. In this example, we consider the following modified Helmholtz equation

$$(\Delta - 10)\, u(x, y, z) = f(x, y, z), \qquad (x, y, z) \in \Omega, \tag{27}$$

$$u(x, y, z) = g(x, y, z), \qquad (x, y, z) \in \partial\Omega. \tag{28}$$

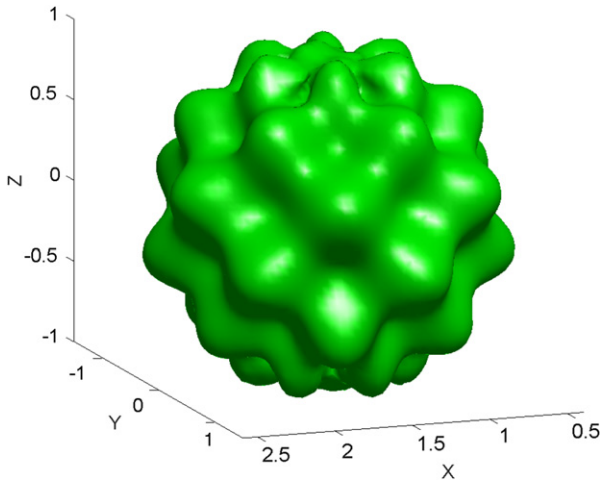

Fig. 7. The profile of a bumpy sphere.

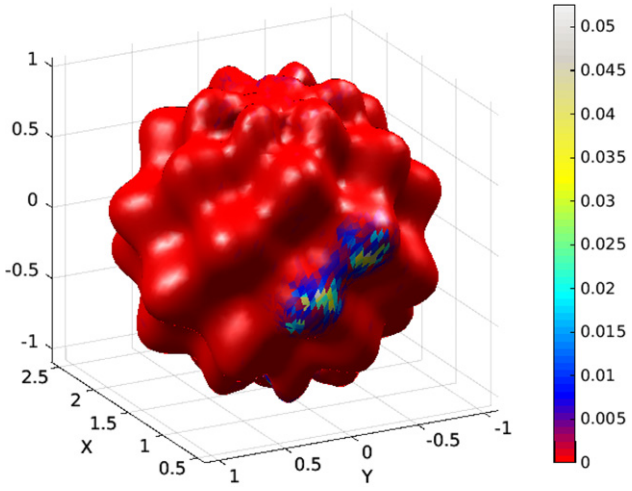

Fig. 8. The profile of absolute relative error on the surface of bumpy sphere with $a = 0.40$.

Table 6
The RMSE, RE and MRAE in Example 2 of the bumpy sphere domain using $n_i = 28, 120$, $n_b = 2500$ and $c = 2.657$ for various values of a.

a	RMSE	RE	MRAE	max $\|f\|$
0.46	9.33E−2	3.95E−2	8.34E−1	256,680
0.45	3.24E−2	1.39E−2	3.82E−1	37,735
0.44	1.45E−2	6.27E−3	2.05E−1	11,326
0.43	7.58E−3	3.31E−3	1.22E−1	4,648
0.42	4.43E−3	1.95E−3	7.94E−2	2,275
0.41	2.80E−3	1.25E−3	5.51E−2	1,247
0.40	1.89E−3	8.51E−4	3.96E−2	740

The functions f and g are given as follows

$$f(x, y) = \frac{2(3a^2 - 3ar + r^2)}{(r - a)^3} - \frac{10r^2}{r - a},$$

$$g(x, y) = \frac{r^2}{r - a},$$

where $r = \sqrt{x^2 + y^2 + z^2}$. The computational domain Ω is the Stanford Bunny [17]. The configuration of the domain is even more challenging than the domain in the previous example. Note that no parametric equation for the surface of the Bunny

Table 7
The RMSE, RE and MRAE in Example 3: using $n_i = 2345$, $n_b = 1889$ and $c = 2.127$ for various values of a.

a	RMSE	RE	MRAE	max $\|f\|$
0.325	4.62E−2	3.48E−1	6.78E−1	10,153
0.320	3.38E−2	2.80E−1	4.76E−1	5,803
0.300	1.27E−2	1.52E−1	1.97E−1	1,110
0.280	5.97E−3	8.67E−2	1.00E−1	342
0.260	3.11E−3	5.16E−2	5.42E−2	134
0.210	7.34E−4	1.52E−2	1.29E−2	24

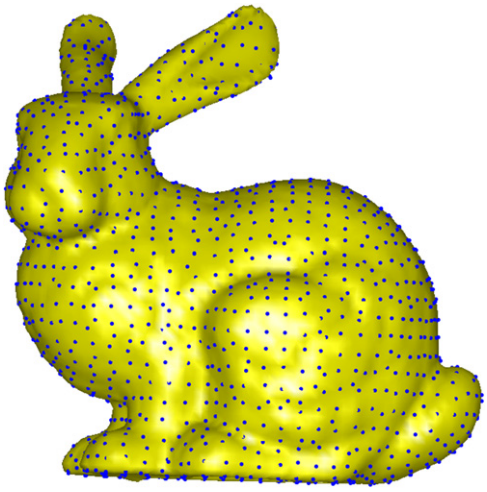

Fig. 9. The profile of Stanford Bunny and the 1889 boundary points.

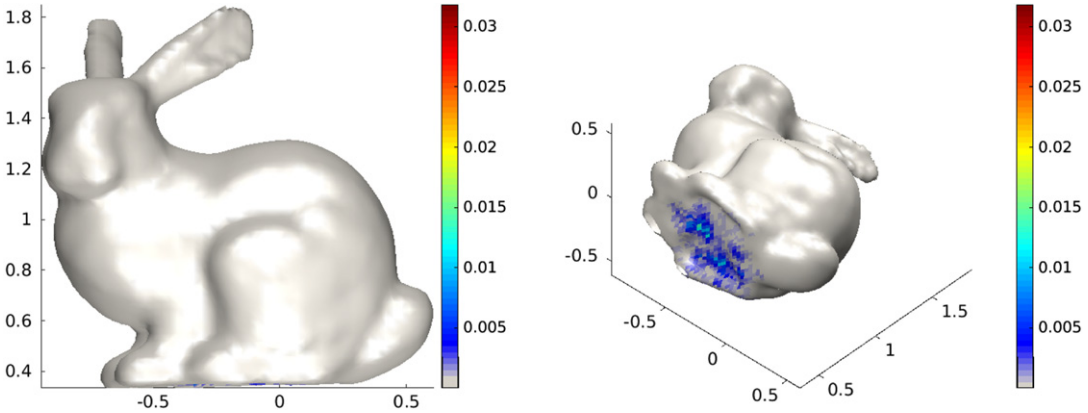

Fig. 10. The profiles of maximum errors on the surface of the Bunny for the case $a = 0.23$.

is available. In the numerical implementation, 1889 scanned cloud data on the boundary surface of the Bunny is available at the website of the Stanford Computer Graphics Laboratory [17]. Since the scale of the original cloud data of Bunny is too small, we magnify these data 10 times larger for the purpose of solving numerical PDEs in a reasonable size of domain. The profile of the Stanford Bunny and its 1889 boundary points is shown in Fig. 9. It is not a trivial task to obtain the interior points so that we can perform our numerical test. To achieve this, we use the technique of the computer graphic in [18] to retrieve 2345 interior points.

To carry out the numerical test using LMAPS, we choose 20 nearest points in the influence domain for each center. In Table 7, we show the numerical accuracy for various a. The shape parameter of MQ is obtained using LOOCV with initial search interval [0,5]. Fig. 10 show the profiles of maximum error on the surface of the Bunny for the case $a = 0.23$. As we can see that the errors at the bottom of the Bunny are much higher than the other part of the surface since this region is closer to the singular point in the vicinity of the outer boundary.

6. Conclusions

In this paper we apply the method of approximate particular solution (LMAPS) [8] to tackle elliptic PDEs with near-singular forcing terms directly without applying a tedious smoothing scheme and enhancing nodes as shown in [11].

In this paper we have improved the original LMAPS by reducing the size of the resultant sparse matrix system. As a result, the LMAPS is further simplified and becomes more efficient. Our numerical results outperform those from CS-RBFs as shown in [11]. Furthermore, the proposed approach can be extended to other differential equations with variable coefficients while the method proposed in [11] can only be applied to solve differential equations where the fundamental solutions are available. We only use a small number of neighboring nodes to localize the PDEs. To our knowledge, none of the global methods, such as the Kansa's method or the method of approximate particular solution (MAPS), can be successful for solving near-singular problems. However, LMAPS, a local method, can achieve reasonable accuracy for near-singular problems.

It is worth mentioning that we also tried to use the local Kansa's method [6] to solve near-singular problems. As far as the performance is concerned, there is no significant difference between these two methods. There are several localized meshless methods available in the literature. But it is not the focus of this paper to compare these methods. However, one conclusion is almost certain that the localized methods are preferred for solving problems with sharp spikes in nonhomogeneous forcing terms.

Acknowledgment

The third author acknowledges the support of the Youth Fund of Taiyuan University of Technology (Project No. 2013T058).

References

[1] E.J. Kansa, Multiquadrics—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.
[2] 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.
[3] 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.
[4] N. Mai-Duy, T. Tran-Cong, Numerical solution of differential equations using multiquadrics radial basis function networks, Internat. J. Numer. Methods Engrg. 23 (2001) 1807–1829.
[5] 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.
[6] C.K. Lee, X. Liu, S.C. Fan, Local multiquadric approximation for solving boundary value problems, Comput. Mech. 30 (2003) 396–409.
[7] 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.
[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] G.E. Fasshauer, J.G. Zhang, On choosing optimal shape parameters for RBF approximation, Numer. Algorithms 45 (2007) 345–368.
[10] 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.
[11] C.S. Chen, G. Kuhn, J. Li, G. Mishuris, Radial basis functions for solving near singular Poisson problems, Comm. Numer. Methods Engrg. 19 (2003) 333–347.
[12] C.-S. Huang, C.F. Lee, A.H.-D. Cheng, Error estimate, optimal shape factor, and high precision computation of multiquadrics collocation method, Eng. Anal. Bound. Elem. 31 (7) (2007) 614–623.
[13] 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.
[14] 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.
[15] Guangming Yao, Local radial basis function methods for solving partial differential equations (Ph.D. thesis) University of Southern Mississippi, 2010.
[16] H.Y. Tian, S. Reutskiy, C.S. Chen, A basis function for approximation and the solution of partial differential equations, Numer. Methods Partial Differential Equations 24 (2008) 1018–1036.
[17] http://graphics.stanford.edu/data/3Dscanrep/.
[18] R. Tankelevich, G. Fairweather, A. Karageorghis, Three-dimensional image reconstruction using the PF/MFS technique, Eng. Anal. Bound. Elem. 33 (2009) 1403–1410.