Engineering Analysis with Boundary Elements 57 (2015) 9–15

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

A local meshless collocation method for solving certain inverse problems

Wen Li [a], Xiaoyan Liu [a,*], Guangming Yao [b]

[a] College of Mathematics, Taiyuan University of Technology, Taiyuan 030024, China
[b] Department of Mathematics, Clarkson University, Potsdam, NY, USA

ARTICLE INFO

Article history:
Received 12 July 2014
Received in revised form
9 November 2014
Accepted 18 November 2014
Available online 26 December 2014

Keywords:
Inverse problem
Compactly supported radial basis functions
Local meshless method
Tikhonov regularization
L-curve

ABSTRACT

In this paper, we propose a meshless scheme based on compactly supported radial basis functions (CS-RBFs) for solving the Cauchy problem of Poisson's equation and the inverse heat conduction problems in 2D. By assuming the unknown boundary condition to be a polynomial function, the inverse problems can be solved using a procedure similar to the process for solving forward problems. We employ Tikhonov regularization technique under L-curve regularization parameter to obtain a stable numerical solution. Numerical results verify the effectiveness and stability of this method.

1. Introduction

Inverse problems arise in scientific, engineering and even medical fields such as non-destructive testing in stress and strain analysis, cardiography, and the heat conduction problem. As we know, these kinds of problems are ill-posed, which means the solutions do not depend continuously on the boundary conditions. Since any small errors caused by the measurement of input data on the boundary or interior of the domain can result in highly amplified errors in the numerical solutions, traditional methods for well-posed forward problems are not suitable for solving inverse problems. Therefore, developments of effective and stable numerical algorithms are essential.

During the last few decades, many numerical methods have been presented for solving inverse problems [1–11]. Among these papers, most of the numerical algorithms are based on the method of fundamental solution (MFS) [1–5,7], the finite difference method (FDM) [6,8], and the finite element method (FEM) [9–11]. However, every method has its own limitations. For FDM and FEM, the cost of generating meshes for three dimensional problems is quite high. Furthermore, the adaptability of FDM to complex domains is poor. Although FEM has better versatility and adaptability to irregular domains, the computational cost in time and space is extremely high for solving large-scale inverse problems. MFS was first applied to solve elliptic boundary value problems by Fairweather and Karageorghis [12]. The pure MFS is limited for solving homogeneous equations when the fundamental solutions are available. Although MFS can be used to solve inhomogeneous problems by combining with the dual reciprocity method (DRM) [13,14], the severe ill-conditioning of the coefficient matrix and the uncertainty for setting the fictitious boundary hinder its application in practical problems.

The meshless methods based on the radial basis functions (RBFs) are of competitive edge, due to their simplicity in selecting interpolation points and high adaptability to domain shape and equation type. Hon and Wu [15] gave the first approach in applying RBFs to solve the Cauchy problem for Laplace equations. Since then, some papers in this area have been published [16–19]. In these papers, the main idea for solving inverse problems is to approximate the solution by a linear combination of RBFs and directly substitute the approximated solution into the governing equation, the boundary conditions, and the over-specified conditions. As we know, the main difficulty in designing an algorithm stems from the ill-posedness of the inverse problem and the ill-conditioning of the coefficient matrix. Furthermore, the condition number of the coefficient matrix increases dramatically with an increase in the number of interpolation points. Therefore, for large-scale problems, where a large number of interpolation nodes are necessary, the coefficient matrix based on the commonly used globally defined RBFs can be dense and highly ill-conditioned. For this reason, compactly supported RBFs (CS-RBFs) which are positive-definite and can result in a sparse matrix are suitable for solving large-scale problems. CS-RBFs have been extensively used for solving forward

* Corresponding author.
E-mail address: lucyyanxiao@163.com (X. Liu).

http://dx.doi.org/10.1016/j.enganabound.2014.11.034
0955-7997/

problems. CS-RBFs have been used in the dual reciprocity boundary element method (DRBEM) for solving Poisson's equation [20–22], Stokes Flow problems [23], in MPS-MFS for solving 3D Hemholtz-type equations [24], and in the collocation method for solving shallow water equations [25]. However, to the best of the authors' knowledge, CS-RBFs have not been used for solving inverse problems yet.

In this paper we propose a stable local meshless numerical method based on CS-RBFs for solving 2D inverse problems with a small number of sensors installed inside the domain. We determine the Dirichlet boundary data on an unreachable boundary. Unlike other direct methods [15–19], we design a novel scheme by first assuming the unknown boundary condition to be a polynomial function and then creating equations based on CS-RBFs in an ingenious process. It is worth mentioning that the size and the number of non-zero elements of the coefficient matrix in the proposed method are much smaller than the traditional direct method using CS-RBFs. Thus, the condition number of the coefficient matrix is significantly smaller.

The paper is organized as follows. In Section 2, we briefly review CS-RBFs. In Section 3, we propose the scheme on how to solve inverse problem for Poisson's equation using CS-RBFs. In Section 4, we propose a 2D IHCP algorithm by following the method presented in Section 3 to further verify the stability of the approach. Furthermore, the Tikhonov regularization method with L-curve scheme is applied to obtain a stable solution. In Section 5, the efficiency and stability of the proposed method are tested in comparison with the conventional direct method used in most papers based on the same CS-RBF.

2. CS-RBFs

Radial basis functions are simple and effective tools in approximating multivariate functions. Let $\mathbf{E}=\{\mathbf{e}_j\}_{j=1}^{l}$ be a set of pairwise distinct points in a domain $\Omega\subseteq\mathbb{R}^2$ with associate values $\{f(\mathbf{e}_j)\}_{j=1}^{l}$. For the commonly used global RBFs φ such as Guassians and multiquadrics, the interpolation matrix $\mathbf{A_E}=\left(\varphi\left(\|\mathbf{e}_k-\mathbf{e}_j\|\right)\right)_{1\le j,k\le l}$ is non-sparse. To obtain a more accurate solution for inverse problems, we want to use as many points as possible when conditions allow. However, for a large number of interpolation points, the condition number of the coefficient matrix based on the global basis function can be quite large, leading to a loss of stability and numerical accuracy. Furthermore, the cost of matrix inverting and storing $\mathbf{A_E}$ could be enormous. To overcome all these difficulties, compactly supported RBFs (CS-RBFs) have been introduced as local basis functions. The construction of the CS-RBFs was first established by Wu [26], followed by Wendland [27], and later by Buhmann [28]. In this paper we will focus on the CS-RBFs constructed by Wendland [27]. These functions are piecewise polynomial with minimal degree in terms of the given order of smoothness. The interpolation matrix $\mathbf{A_E}$ based on CS-RBF is sparse and positive definite. A list of 2D CS-RBFs is given in Table 1. In this table, the cut-off function $(r)_+$ is defined to be r if $r\ge 0$ and to be zero elsewhere.

In Table 1, the radius of the support of the function has been normalized to 1. In the real application, we can re-scale the function in this table with the support of radius α using $\varphi(r/\alpha)$ for $\alpha>0$. The sparseness of the interpolation matrix $\mathbf{A_E}$ can be suitably adjusted by choosing the scaling factor α. If α is too small, the reproduction quality is poor, while if α is too large, the matrix $\mathbf{A_E}$ is no longer sparse and it will lost its attractiveness in real applications. Hence, a reasonable choice of the scaling factor α is crucial to compromise between the stability and quality of the approximation.

Table 1
Wendland's CS-RBFs in 2D.

$(1-r)_+^2\in C^0$
$(1-r)_+^4(4r+1)\in C^2$
$(1-r)_+^6(35r^2+18r+3)\in C^4$

3. The local meshless method for a stationary inverse heat conduction equation

First, consider the following inverse problem for Poisson's equation:

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

$$u(\mathbf{x})=g(\mathbf{x}),\quad \mathbf{x}\in\partial\Omega_1, \tag{2}$$

$$u(\mathbf{x})=h(\mathbf{x}),\quad \mathbf{x}\in\partial\Omega_2, \tag{3}$$

where $\Omega\subseteq\mathbb{R}^2$ is a bounded domain with boundary $\partial\Omega$, and $\partial\Omega=\partial\Omega_1\cup\partial\Omega_2$, $\partial\Omega_1\cap\partial\Omega_2=\varnothing$ and $\partial\Omega_2\neq\varnothing$. Δ is the Laplacian, f and g are given functions, and h is unknown. Note that $\mathbf{x}=(x,y)$. In addition, the over-specified condition is given as follows:

$$u(\mathbf{x}_i^*)=q(\mathbf{x}_i^*),\quad \mathbf{x}_i^*\in\Omega,\quad i=1,2,\ldots,nq, \tag{4}$$

where $\mathbf{x}_i^*$, $i=1,2,\ldots,nq$, are certain interior points at which sensors are fixed in the reachable part of the domain. Therefore, the measured values $q(\mathbf{x}_i^*)$ $(i=1,2,\ldots,nq)$ are known.

In the inverse problem described above, u at any points on boundary $\partial\Omega_2$ and in Ω should be determined. In Section 3, the main idea and process of applying a local meshless method based on CS-RBF to solve this kind of problem will be explained.

Here, we use a polynomial of degree d to approximate the Dirichlet boundary condition on $\partial\Omega_2$, which means

$$h(\mathbf{x})\simeq\hat{h}(\mathbf{x})=b_0+b_1x+b_2y+b_3x^2+b_4xy+b_5y^2+\cdots+b_{d(d+3)/2}y^d. \tag{5}$$

The above polynomial can be represented in vector form:

$$\hat{h}(\mathbf{x})=\mathbf{p}(\mathbf{x})\mathbf{b},\quad \mathbf{x}\in\partial\Omega_2 \tag{6}$$

where

$$\mathbf{p}(\mathbf{x})=(1\ \ x\ \ y\ \ x^2\ \ xy\ \ y^2\ \ \cdots\ \ y^d)$$
$$\mathbf{b}=\left(b_0\ \ b_1\ \ \cdots\ \ b_{d(d+3)/2}\right)^T \tag{7}$$

In order to solve problems (1)–(4), we tentatively assume that $h(\mathbf{x})$ is known as shown in (6), which means $\mathbf{b}$ in (6) is given. Then problems (1)–(4) can be taken as the well-posed forward problem. We apply Kansa's method [29] based on compactly supported RBFs to solve this hypothetical forward problem. We proceed in the following way.

Let $\{\mathbf{x}_j\}_{j=1}^{n}$ be a set of uniformly distributed pairwise distinct interpolation points in $\Omega\cup\partial\Omega$. Note that $\{\mathbf{x}_j\}_{j=1}^{ni}\subseteq\Omega$, $\{\mathbf{x}_j\}_{j=ni+1}^{ni+nb1}\subseteq\partial\Omega_1$, and $\{\mathbf{x}_j\}_{j=ni+nb1+1}^{n}\subseteq\partial\Omega_2$, and $n=ni+nb1+nb2$. $\|\cdot\|$ denotes the Euclidean norm. Then we seek to approximate u by $\hat{u}$ as follows:

$$u(\mathbf{x})\simeq\hat{u}(\mathbf{x})=\sum_{i=1}^{n}a_j\varphi_\alpha(r_j),\quad \mathbf{x}\in\Omega\cup\partial\Omega, \tag{8}$$

where $r_j=\|\mathbf{x}-\mathbf{x}_j\|$, and $\varphi_\alpha(r_j)=\varphi(r_j/\alpha)$ is a CS-RBF with scaling factor α.

According to Kansa's method [29], the coefficient $\{a_j\}_{j=1}^{n}$ can be obtained by the collocation approach

$$\sum_{j=1}^{n}a_j\Delta\varphi_\alpha\left(\|\mathbf{x}_i-\mathbf{x}_j\|\right)=f(\mathbf{x}_i),\quad 1\le i\le ni, \tag{9}$$

$$\sum_{j=1}^{n}a_j\varphi_\alpha\left(\|\mathbf{x}_i-\mathbf{x}_j\|\right)=g(\mathbf{x}_i),\quad ni+1\le i\le ni+nb1, \tag{10}$$

$$\sum_{j=1}^{n} a_j \varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|)=\hat{h}(\mathbf{x}_i),\quad ni+nb1+1\le i\le ni+nb1+nb2. \tag{11}$$

The system of Eqs. (9)–(11) can be formulated in the following block matrix form:

$$\begin{pmatrix}\Delta\mathbf{\Phi_I}\\ \mathbf{\Phi_{B1}}\\ \mathbf{\Phi_{B2}}\end{pmatrix}\mathbf{a}=\begin{pmatrix}\mathbf{f}\\ \mathbf{g}\\ \hat{\mathbf{h}}\end{pmatrix} \tag{12}$$

where

$$\Delta\mathbf{\Phi_I}=\left(\Delta\varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|)\right)_{1\le i\le ni,1\le j\le n}$$
$$\mathbf{\Phi_{B1}}=\left(\varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|)\right)_{ni+1\le i\le ni+nb1,1\le j\le n}$$
$$\mathbf{\Phi_{B2}}=\left(\varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|)\right)_{ni+nb1+1\le i\le n,1\le j\le n}$$

and

$$\mathbf{a}=\begin{pmatrix}a_1\\ a_2\\ \vdots\\ a_n\end{pmatrix},\quad \mathbf{f}=\begin{pmatrix}f(\mathbf{x}_1)\\ f(\mathbf{x}_2)\\ \vdots\\ f(\mathbf{x}_{ni})\end{pmatrix},\quad \mathbf{g}=\begin{pmatrix}g(\mathbf{x}_{ni+1})\\ g(\mathbf{x}_{ni+2})\\ \vdots\\ g(\mathbf{x}_{ni+nb1})\end{pmatrix},\quad \hat{\mathbf{h}}=\begin{pmatrix}\hat{h}(\mathbf{x}_{ni+nb1+1})\\ \hat{h}(\mathbf{x}_{ni+nb1+2})\\ \vdots\\ \hat{h}(\mathbf{x}_n)\end{pmatrix}.$$

From (6), we have

$$\begin{pmatrix}\mathbf{f}\\ \mathbf{g}\\ \hat{\mathbf{h}}\end{pmatrix}=\begin{pmatrix}\mathbf{0}_{ni\times\frac{(d+1)(d+2)}{2}} & \mathbf{f}\\ \mathbf{0}_{nb1\times\frac{(d+1)(d+2)}{2}} & \mathbf{g}\\ \mathbf{P} & \mathbf{0}_{nb2\times1}\end{pmatrix}\begin{pmatrix}\mathbf{b}\\ 1\end{pmatrix} \tag{13}$$

where

$$\mathbf{P}=\begin{pmatrix}\mathbf{p}(\mathbf{x}_{ni+nb1+1})\\ \mathbf{p}(\mathbf{x}_{ni+nb1+2})\\ \vdots\\ \mathbf{p}(\mathbf{x}_n)\end{pmatrix}.$$

We denote

$$\mathbf{\Phi}=\begin{pmatrix}\Delta\mathbf{\Phi_I}\\ \mathbf{\Phi_{B1}}\\ \mathbf{\Phi_{B2}}\end{pmatrix},\quad \begin{pmatrix}\mathbf{H_1} & \mathbf{h_f}\end{pmatrix}=\begin{pmatrix}\mathbf{0}_{ni\times\frac{(d+1)(d+2)}{2}} & \mathbf{f}\\ \mathbf{0}_{nb1\times\frac{(d+1)(d+2)}{2}} & \mathbf{g}\\ \mathbf{P} & \mathbf{0}_{nb2\times1}\end{pmatrix}.$$

Then, from (13), (12) becomes

$$\mathbf{\Phi a}=\begin{pmatrix}\mathbf{H_1} & \mathbf{h_f}\end{pmatrix}\begin{pmatrix}\mathbf{b}\\ 1\end{pmatrix}. \tag{14}$$

Consequently, we can obtain $\mathbf{a}$ by

$$\mathbf{a}=\mathbf{\Phi}^{-1}\begin{pmatrix}\mathbf{H_1} & \mathbf{h_f}\end{pmatrix}\begin{pmatrix}\mathbf{b}\\ 1\end{pmatrix}=\mathbf{\Phi}^{-1}\mathbf{H_1 b}+\mathbf{\Phi}^{-1}\mathbf{h_f}. \tag{15}$$

However, $\mathbf{b}$ in (6) is unknown. Once $\mathbf{b}$ is obtained, the Dirichlet boundary data at any point on $\partial\Omega_2$ can be approximated by (6) and the coefficient vector $\mathbf{a}$ can be calculated by (15). Then, we can also approximate the value of u at any node in Ω by (8). Therefore, it is crucial to determine $\mathbf{b}$.

Next, we approximate u by $\hat{u}$ as shown in (8) by fitting the given data set $\{\mathbf{x}_i^*\}_{i=1}^{nq}$ in domain Ω with the imposed conditions (4):

$$u(\mathbf{x}_i^*)\simeq\hat{u}(\mathbf{x}_i^*)=\sum_{j=1}^{n} a_i\varphi_\alpha(\|\mathbf{x}_i^*-\mathbf{x}_j\|)=q(\mathbf{x}_i^*),\quad i=1,2,\ldots,nq. \tag{16}$$

It follows that (16) can be reformulated as

$$\mathbf{\Phi}^*\mathbf{a}=\mathbf{q} \tag{17}$$

where

$$\mathbf{\Phi}^*=\left(\varphi_\alpha(\|\mathbf{x}_i^*-\mathbf{x}_j\|)\right)_{1\le i\le nq,1\le j\le n}$$

$$\mathbf{q}=\left(q(\mathbf{x}_1^*)\ q(\mathbf{x}_2^*)\ \cdots\ q(\mathbf{x}_{nq}^*)\right)^T.$$

From (15) and (17), we have

$$\mathbf{\Phi}^*(\mathbf{\Phi}^{-1}\mathbf{H_1 b}+\mathbf{\Phi}^{-1}\mathbf{h_f})=\mathbf{\Phi}^*\mathbf{\Phi}^{-1}\mathbf{H_1 b}+\mathbf{\Phi}^*\mathbf{\Phi}^{-1}\mathbf{h_f}=\mathbf{q}. \tag{18}$$

Denote $\mathbf{A_1}=\mathbf{\Phi}^*\mathbf{\Phi}^{-1}\mathbf{H_1}$, $\mathbf{h_2}=\mathbf{\Phi}^*\mathbf{\Phi}^{-1}\mathbf{h_f}$, (18) can be written as

$$\mathbf{A_1 b}+\mathbf{h_2}=\mathbf{q} \tag{19}$$

which is equivalent to

$$\mathbf{A_1 b}=\mathbf{q}-\mathbf{h_2}. \tag{20}$$

Due to the ill-posedness of the original problem, we need to adopt a regularization method to solve the linear system (20). In this paper, we use the Tikhonov regularization technique [30] because this method has been proved very effective compared with other techniques for solving ill-posed problems [31] and widely used in meshless methods [4,5,19]. Some other outstanding regularization approaches committed to get stable numerical solutions have also been developed [31,32], but they will not be further discussed in this paper. The Tikhonov regularized solution for (20) can be obtained by solving the following minimization problem:

$$\min_{\mathbf{b}}\{\|\mathbf{A_1 b}-(\mathbf{q}-\mathbf{h_2})\|^2+\beta^2\|\mathbf{b}\|^2\} \tag{21}$$

where the regularization parameter β is to be determined. The L-curve method is suitable for both the square matrix and non-square matrix cases [31]. Since the coefficient matrix in (20) might be non-square, we adopt the L-curve method to get an appropriate value of β. The L-curve method was employed by Lawson and Hanson [33] in 1974 and by Hansen and O'Leary [34] in 1993.

As mentioned above, once $\mathbf{b}$ is determined, the value of u at any point on $\partial\Omega_2$ and in Ω can be approximated by using (6), (8), and (15).

4. The local meshless method for an IHCP

The inverse heat conduction problem (IHCP) is a typical time dependent inverse problem. In this section, we consider solving a 2D IHCP using the procedure formulated in the last section. Consider the IHCP as follows:

$$\frac{\partial}{\partial t}u(\mathbf{x},t)=\Delta u(\mathbf{x},t),\quad \mathbf{x}\in\Omega,\ 0\le t\le T, \tag{22}$$

$$u(\mathbf{x},0)=f_t(\mathbf{x}),\quad \mathbf{x}\in\Omega\cup\partial\Omega, \tag{23}$$

$$u(\mathbf{x},t)=g_t(\mathbf{x},t),\quad \mathbf{x}\in\partial\Omega_1,\ 0\le t\le T, \tag{24}$$

$$u(\mathbf{x},t)=h_t(\mathbf{x},t),\quad \mathbf{x}\in\partial\Omega_2,\ 0\le t\le T, \tag{25}$$

where T denotes the terminal time, and f_t and g_t are given functions, and h_t is unknown. $\Omega\subseteq\mathbb{R}^2$ is a bounded domain with boundary $\partial\Omega=\partial\Omega_1\cup\partial\Omega_2$, $\partial\Omega_1\cap\partial\Omega_2=\varnothing$.

The additional specification is given as follows:

$$u(\mathbf{x}_i^*,t)=q_t(\mathbf{x}_i^*,t),\quad \mathbf{x}_i^*\in\Omega,\ i=1,2,\ldots,nq, \tag{26}$$

where $\mathbf{x}_i^*$ $(i=1,2,\ldots,nq)$ are certain interior points at which sensors are installed so that the temperature can be measured.

To solve this problem, the standard Euler difference scheme is used to discretize the time variable. We choose time step size $\tau=T/M$. Then (22)–(26) can be discretized as

$$\frac{u^{m+1}(\mathbf{x})-u^m(\mathbf{x})}{\tau}=\Delta u^{m+1}(\mathbf{x}),\quad \mathbf{x}\in\Omega,\ m=0,1,\ldots,M-1, \tag{27}$$

$$u^0(\mathbf{x})=f_t(\mathbf{x}),\quad \mathbf{x}\in\Omega\cup\partial\Omega, \tag{28}$$

$$u^{m+1}(\mathbf{x})=g_t^{m+1}(\mathbf{x}),\quad \mathbf{x}\in\partial\Omega_1,\ m=0,1,\ldots,M-1, \tag{29}$$

$$u^{m+1}(\mathbf{x}) = h_t^{m+1}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega_2, \quad m = 0, 1, \ldots, M-1, \tag{30}$$

and

$$u^{m+1}(\mathbf{x}_i^*) = q_t^{m+1}(\mathbf{x}_i^*), \quad \mathbf{x}_i^* \in \Omega, \quad i = 1, 2, \ldots, nq, \quad m = 0, 1, \ldots, M-1. \tag{31}$$

We rewrite (27) as follows:

$$u^{m+1}(\mathbf{x}) - \tau\Delta u^{m+1}(\mathbf{x}) = u^m(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad m = 0, 1, \ldots, M-1, \tag{32}$$

where the superscript $m+1$ means function values at time $t = (m+1)\tau$.

Similar to the process in Section 3, we approximate u^{m+1} by $\hat{u}^{m+1}$ in the form of a linear combination of CS-RBFs as shown in (8):

$$u^{m+1}(\mathbf{x}) \simeq \hat{u}^{m+1}(\mathbf{x}) = \sum_{j=1}^{n} a_j^{m+1} \varphi_\alpha(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \Omega, \quad m = 0, 1, \ldots, M-1, \tag{33}$$

and h_t^{m+1} by $\hat{h}_t^{m+1}$ as (6)

$$h_t^{m+1}(\mathbf{x}) \simeq \hat{h}_t^{m+1}(\mathbf{x}) = \mathbf{p}(\mathbf{x})\mathbf{b}^{m+1}, \quad \mathbf{x} \in \partial\Omega_2. \tag{34}$$

The idea for solving this time-dependent problem is similar to the time-independent problem described in the last section. We just need to rewrite (12) as follows:

$$\begin{pmatrix} \mathbf{\Phi_I} - \tau\Delta\mathbf{\Phi_I} \\ \mathbf{\Phi_{B1}} \\ \mathbf{\Phi_{B2}} \end{pmatrix} \mathbf{a}^{m+1} = \begin{pmatrix} \mathbf{u}^m \\ \mathbf{g}_\mathbf{t}^{m+1} \\ \hat{\mathbf{h}}_\mathbf{t}^{m+1} \end{pmatrix} \tag{35}$$

where $\mathbf{u}^m = (u^m(\mathbf{x}_1)\ u^m(\mathbf{x}_2)\ \cdots\ u^m(\mathbf{x}_{ni}))^T$. Hence, (15) becomes

$$\mathbf{a}^{m+1} = \mathbf{\Phi}_\mathbf{t}^{-1}\mathbf{H_1}\mathbf{b}^{m+1} + \mathbf{\Phi}_\mathbf{t}^{-1}\mathbf{h}_\mathbf{u}^{m+1} \tag{36}$$

where

$$\mathbf{\Phi_t} = \begin{pmatrix} \mathbf{\Phi_I} - \tau\Delta\mathbf{\Phi_I} \\ \mathbf{\Phi_{B1}} \\ \mathbf{\Phi_{B2}} \end{pmatrix}, \quad \mathbf{H_1} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{P} \end{pmatrix}, \quad \mathbf{h}_\mathbf{u}^{m+1} = \begin{pmatrix} \mathbf{u}^m \\ \mathbf{g}_\mathbf{t}^{m+1} \\ \mathbf{0} \end{pmatrix}.$$

Similarly, we can obtain coefficient vector $\mathbf{b}^{m+1}$ at $t = (m+1)\tau$ by solving

$$\mathbf{A_{1t}}\mathbf{b}^{m+1} = \mathbf{q}_\mathbf{t}^{m+1} - \mathbf{h}_\mathbf{2}^{m+1} \tag{37}$$

where $\mathbf{A_{1t}} = \mathbf{\Phi}^*\mathbf{\Phi}_\mathbf{t}^{-1}\mathbf{H_1}$, $\mathbf{h}_\mathbf{2}^{m+1} = \mathbf{\Phi}^*\mathbf{\Phi}_\mathbf{t}^{-1}\mathbf{h}_\mathbf{u}^{m+1}$ and $\mathbf{q}_\mathbf{t}^{m+1} = (q_t^{m+1}(\mathbf{x}_1^*)\ q_t^{m+1}(\mathbf{x}_2^*)\ \cdots\ q_t^{m+1}(\mathbf{x}_{nq}^*))^T$.

To be more specific, we describe the numerical procedure for solving this IHCP as the iteration process as follows:

(a) Let $m=0$, which means the initial time $t=0$. Since $\mathbf{u}^0 = (u^0(\mathbf{x}_1)\ u^0(\mathbf{x}_2)\ \cdots\ u^0(\mathbf{x}_{ni}))^T$ has been given, $\mathbf{h}_\mathbf{u}^{m+1}$ is known.
(b) Using the Tikhonov regularization technique with L-curve to solve (37), we can obtain $\mathbf{b}^{m+1}$.
(c) Obtain an approximate temperature on $\partial\Omega_2$ at time $t = (m+1)\tau$ by (34).
(d) Calculate coefficient vector $\mathbf{a}^{m+1} = (a_1^{m+1}\ a_2^{m+1}\ \cdots\ a_n^{m+1})^T$ by (36), and get $\mathbf{u}^{m+1}$ at time $t = (m+1)\tau$ by (33).
(e) Let $m = m+1$ and repeat (b), (c) and (d) until $m = M-1$.

From the procedure described above, we can obtain the approximate temperature in Ω and on $\partial\Omega_2$ at every time node $t = (m+1)\tau$, $m = 0, 1, \ldots, M-1$.

5. Numerical examples

To validate the efficacy of the proposed method, we consider three examples in this section. The computations were performed using MATLAB on an Intel Core i5 in Windows 7 Home Basic 64 bit. The relative error and the root-mean-square-error (RMSE) will be used in this section to demonstrate the accuracy of the approximated solutions:

$$E_r(\mathbf{u}) = \frac{\|\mathbf{u} - \hat{\mathbf{u}}\|}{\|\mathbf{u}\|},$$

and

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(\hat{u}(\mathbf{x}_j) - u(\mathbf{x}_j))^2}$$

where $\mathbf{u} = (u_1\ u_2\ \cdots\ u_N)^T$ and $\hat{\mathbf{u}} = (\hat{u}_1\ \hat{u}_2\ \cdots\ \hat{u}_N)^T$ denote the numerical solution and the analytical solution respectively at these points, and N is the number of test points.

With the increase of the smoothness of CS-RBF and the degree d of the polynomial in (5), the accuracy of solution can be slightly improved and the CPU time will be slightly longer [20]. However, as we know, high-degree polynomial interpolation may arise the numerical oscillation problem. In this paper, we choose basis function with moderate smoothness and degree. CS-RBF $(1-r/\alpha)_+^6(35(r/\alpha)^2 + 18r/\ \alpha + 3)$ is used in these examples, and the degree of polynomial $d=5$ in first two examples, $d=4$ in Example 3. The effectiveness and efficiency of the method are also greatly depended on the scaling factor α. Generally, the larger the value of α, the more accurate the interpolation, whereas the lower the efficiency because of the higher density of the matrix. On the other hand, we should note that numerical solution can sometimes be unstable after a certain limit of α [35]. Therefore, in this paper, the values of α we choose are neither too small nor too large. Moreover, the more the interpolation points, the smaller the value of α.

For time independent problems, we consider the problem in a square $\Omega = [0, 1] \times [0, 1]$ given in Fig. 1 with an unknown boundary condition on $\partial\Omega_2$.

To verify the effectiveness of the method proposed in this paper, we compare our method with the conventional direct method [16,17]. For fairness of comparison, we also choose the same type of CS-RBF as the basis function in the following direct method. The process of the conventional direct method is briefly described as follows.

To solve the inverse problem (1)–(4) using the conventional direct method, we approximate u by $\tilde{u}$ in the form of

$$u(\mathbf{x}) \simeq \tilde{u}(\mathbf{x}) = \sum_{j=1}^{nk} c_j \varphi_\alpha(r_j), \quad \mathbf{x} \in \Omega \cup \partial\Omega, \tag{38}$$

where $nk = ni + nb1 + nq$, $r_j = \|\mathbf{x} - \mathbf{x}_j\|$, $\varphi_\alpha(r_j)$ is a CS-RBF with scaling factor α. $\{\mathbf{x}_j\}_{j=1}^{nk} = \{\mathbf{x}_j\}_{j=1}^{ni} \cup \{\mathbf{x}_j\}_{j=ni+1}^{ni+nb1} \cup \{\mathbf{x}_j\}_{j=ni+nb1+1}^{nk}$,

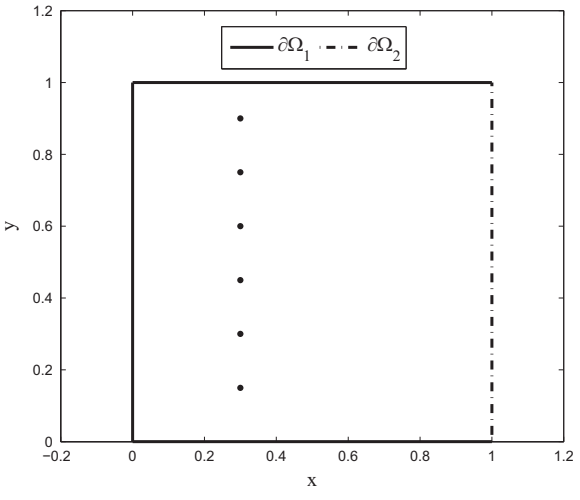

Fig. 1. The square domain with boundary $\partial\Omega_2$. The dots in the domain are for over-specified conditions.

where $\{\mathbf{x}_j\}_{j=1}^{ni} \subseteq \Omega$, $\{\mathbf{x}_j\}_{j=ni+1}^{ni+nb1} \subseteq \partial\Omega_1$, and $\{\mathbf{x}_j\}_{j=ni+nb1+1}^{nk} = \{\mathbf{x}_j^*\}_{j=1}^{nq} \subseteq \Omega$.

Note that the coefficient $\{c_j\}_{j=1}^{nk}$ can be obtained by satisfying the following conditions:

$$\sum_{j=1}^{nk} c_j \Delta\varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|) = f(\mathbf{x}_i), \quad 1 \le i \le ni, \tag{39}$$

$$\sum_{j=1}^{nk} c_j \varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|) = g(\mathbf{x}_i), \quad ni+1 \le i \le ni+nb1, \tag{40}$$

$$\sum_{j=1}^{nk} c_j \varphi_\alpha(\|\mathbf{x}_i^*-\mathbf{x}_j\|) = q(\mathbf{x}_i^*), \quad 1 \le i \le nq. \tag{41}$$

For our simplicity, (39)–(41) can be written in the following matrix form:

$$\mathbf{\Phi_k c} = \mathbf{f_k}, \tag{42}$$

where

$$\mathbf{\Phi_k} = \begin{pmatrix} \Delta\varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|)_{1\le i\le ni,1\le j\le nk} \\ \varphi_\alpha(\|\mathbf{x}_i-\mathbf{x}_j\|)_{ni+1\le i\le ni+nb1,1\le j\le nk} \\ \varphi_\alpha(\|\mathbf{x}_i^*-\mathbf{x}_j\|)_{1\le i\le np,1\le j\le nk} \end{pmatrix},$$

$$\mathbf{c} = (c_1 \ c_2 \ \cdots \ c_{nk})^T,$$

$$\mathbf{f_k} = \left(f(\mathbf{x}_1) \ \cdots \ f(\mathbf{x}_{ni}) \ g(\mathbf{x}_{ni+1}) \ \cdots \ g(\mathbf{x}_{ni+nb1}) \ q(\mathbf{x}_1^*) \ \cdots \ q(\mathbf{x}_{nq}^*)\right)^T.$$

The system (42) can be solved by applying the Tikhonov regularization technique and L-curve method. Consequently, the value of u at any point on $\partial\Omega_2$ and in Ω can be approximated by (38). We can see that the above direct method involves only one simple step, which is identical to the RBF meshless collocation method for solving the well-posed problems.

Example 1. We first consider the stationary inverse heat conduction problem. In this example, the analytical solution for problems (1)–(4) is known as

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

All the discrete value of $f(x,y)$, $g(x,y)$ and $q(x,y)$ can be computed from the above solution. The Dirichlet boundary condition on $\partial\Omega_2 = \{(x,y) : x = 1, 0 \le y \le 1\}$ is to be determined. The interpolation points are uniformly distributed inside the square and on the boundary. As shown in Fig. 1, $\partial\Omega_1 = \partial\Omega \setminus \Omega_2$. Furthermore, there are $nq=6$ sensors evenly installed on the line $x=0.3$.

Since the points on $\partial\Omega_2$ have the same abscissa values, (5) can be reduced to

$$\hat{h}(\mathbf{x}) = b_0 + b_1 y + b_2 y^2 + \cdots + b_d y^d, \tag{43}$$

where $d=5$ in this example.

In practice, measurement error should be taken into account. We use noisy data $\tilde{q}(\mathbf{x}_i^*) = q(\mathbf{x}_i^*) + \sigma * \mathrm{rand}(i)$ in this example, where $q(\mathbf{x}_i^*)$ is the exact data, magnitude σ indicates the error level and $\mathrm{rand}(i)$ is a random number in $[-1, 1]$. Here, we set $\sigma = 0.1$.

The relative errors and RMSE of our approximation u on $\partial\Omega_2$ with different numbers of interpolation points and radii α using the traditional method and the proposed method are shown in Table 2, in which CT denotes the CPU time and nz is the number of non-zero elements of the coefficient matrix. In this example, the optimal regularization parameters β obtained by the L-curve method are also provided.

The errors stem not only from the scheme itself, but also from the noise added to $q(\mathbf{x}_i^*)$ and the L-curve method used in the paper. Using the conventional direct method based on CS-RBF, numerical results have large errors. This is fundamentally due to the ill-posedness of the inverse problem and the instability of the conventional direct method. However, under the same conditions, referring to the same points, the radial basis function and parameters, the relative error and RMSE on $\partial\Omega_2$ by using the method we proposed are superior to the conventional method. Taking into account the randomness of disturbance added to $q(\mathbf{x}_i^*)$, the errors using our proposed method are basically improved with the increase of the number of interpolation nodes and scaling factor α.

In Table 2, we also show that the computational efficiency of our proposed method is significantly superior to that of the conventional direct method. The MATLAB codes of Tikhonov regularization and L-curve method we used are developed by Hansen [36], and these codes are based on singular value decomposition (SVD). In these experiments, we found that, for both the conventional method and the proposed method, the process of singular value decomposition accounted for the vast majority of the running time of the whole process. Therefore, the efficiency of each algorithm is greatly affected by the size and the number of non-zero elements of the coefficient matrix used for decomposition. The size of the coefficient matrix $\mathbf{A_1}$ of (20) in the proposed method is $nq \times (d+1)(d+2)/2$. Note that, in this example, the size of $\mathbf{A_1}$ is $nq \times (d+1)$, which is 6×6, since we use (43) instead of (5). Therefore, no matter how many nodes we select, the number of non-zero elements of $\mathbf{A_1}$ is always 36. Whereas the size of the coefficient matrix $\mathbf{\Phi_k}$ of the system (42) in the conventional direct method is $nk \times nk$ which are 1646×1646, 4046×4046 and $10\,106 \times 10\,106$ corresponding to $n=1681$, 4141, and 10 201 respectively. The size and the number of non-zero elements of $\mathbf{\Phi_k}$ are much larger than that of $\mathbf{A_1}$, and thus the proposed method

Table 2
Relative error and RMSE of u on $\partial\Omega_2$.

n	α	Conventional direct method					Proposed method				
		E_r	RMSE	CT (s)	nz	β	E_r	RMSE	CT (s)	nz	β
1681	0.2	0.81	3.63	34.87	270 128	0.0819	0.057	0.25	1.78	36	0.0205
	0.3	0.73	3.25	35.40	554 492	0.0370	0.049	0.22	2.75	36	0.0277
	0.4	0.65	2.91	39.90	907 104	0.0192	0.048	0.21	3.98	36	0.0385
4141	0.2	0.82	3.69	552.83	1 680 006	0.0096	0.039	0.18	6.17	36	0.0059
	0.3	0.57	2.53	607.56	3 452 316	0.0048	0.058	0.26	9.12	36	0.0359
	0.4	0.65	2.91	639.86	5 572 438	0.0023	0.061	0.27	16.04	36	0.0117
10 201	0.1	Out of memory					0.082	0.36	12.99	36	0.0388
	0.15	Out of memory					0.059	0.26	20.22	36	0.0334
	0.2	Out of memory					0.046	0.20	37.62	36	0.0219

Table 3
Relative error and RMSE of u on $\partial\Omega_2$.

n	α	Conventional direct method					Proposed method				
		E_r	RMSE	CT (s)	nz	β	E_r	RMSE	CT (s)	nz	β
1681	0.2	0.83	2.12	34.52	270 116	0.0417	0.077	0.198	1.93	36	0.0460
	0.3	0.75	1.94	37.92	554 492	0.0405	0.030	0.076	3.52	36	0.0667
	0.4	0.69	1.79	41.03	907 618	0.0435	0.044	0.112	4.08	36	0.0403
4141	0.2	0.83	2.14	551.38	1 679 994	0.0036	0.035	0.089	6.94	36	0.0497
	0.3	0.72	1.85	563.02	3 452 316	0.0035	0.022	0.057	10.21	36	0.0452
	0.4	0.69	1.79	567.82	5 573 720	0.0061	0.038	0.097	12.11	36	0.0602
10 201	0.1	Out of memory					0.043	0.111	13.53	36	0.0527
	0.15	Out of memory					0.039	0.100	25.43	36	0.0600
	0.2	Out of memory					0.033	0.085	37.15	36	0.0700

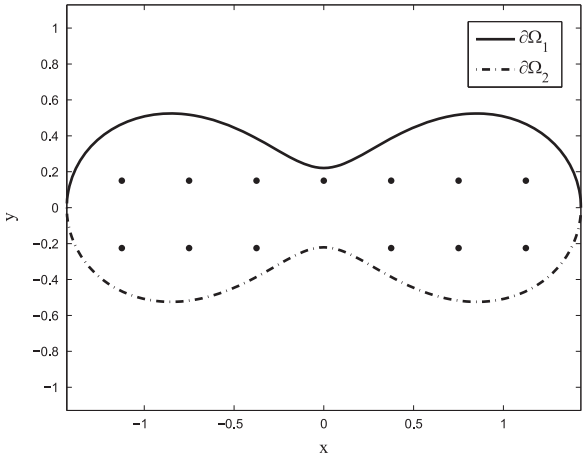

Fig. 2. Peanut domain boundaries. The nodes inside the domain are for over-specified conditions. $\partial\Omega_2$ is represented by a dotted line.

Table 4
Relative error and RMSE of u.

t	E_r on $\partial\Omega_2$	RMSE on $\partial\Omega_2$	E_r in Ω	RMSE in Ω
0.1	3.15E−02	7.68E−02	1.24E−02	2.72E−02
0.2	2.22E−02	6.03E−02	7.7E−03	1.89E−02
0.3	2.87E−02	8.66E−02	9.5E−03	2.63E−02
0.4	1.61E−02	5.38E−02	5.2E−03	1.61E−02
0.5	1.78E−02	6.50E−02	7.1E−03	2.41E−02
0.6	1.73E−02	6.91E−02	6.5E−03	2.43E−02
0.7	1.52E−02	6.58E−02	5.5E−03	2.23E−02
0.8	1.97E−02	9.22E−02	6.5E−03	2.88E−02
0.9	1.41E−02	7.11E−02	4.9E−03	2.32E−02
1	2.06E−02	1.10E−01	7.5E−03	3.87E−02

can dramatically improve the computational efficiency and the condition number of the coefficient matrix.

Example 2. In the second example, we consider the same type of problem as shown in Example 1. $f(x,y)$, $g(x,y)$ and $q(x,y)$ in problems (1)–(4) are given based on the exact solution

$$u(x,y)=x^2+2y+e^{-y}\sin(x), \quad (x,y)\in\Omega\cup\partial\Omega.$$

In this example, six sensors are evenly placed along the line $x=0.4$ in the square domain. Other parameters in Example 1 and Example 2 are the same. Numerical results are given in Table 3.

To further test the proposed method, we add noise to both $g(\mathbf{x}_i)$ and $q(\mathbf{x}_i^*)$. This implies $\tilde{g}(\mathbf{x}_i)=g(\mathbf{x}_i)+\sigma*\mathrm{rand}(i)$ and $\tilde{q}(\mathbf{x}_i^*)=q(\mathbf{x}_i^*)+\sigma*\mathrm{rand}(i)$. Here, let $\sigma=0.1$. From Table 3, we can see that the numerical results are still reasonable and stable.

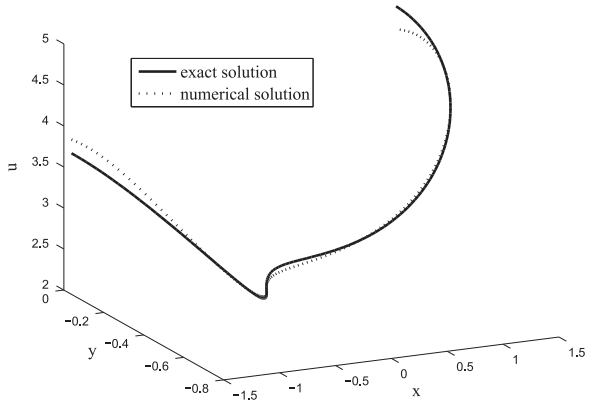

Fig. 3. Numerical (dots) and exact (curve) temperatures on $\partial\Omega_2$ at $t=1$.

Example 3. For the time dependent example, we consider an irregular domain as shown in Fig. 2.

The exact solution of (22)–(26) is given by

$$u(x,y,t)=x^2+y^2+e^{-t}\sin(x)+e^{-t}\cos(y)+4t, \quad (x,y)\in\Omega\cup\partial\Omega, \quad 0\le t\le 1. \tag{44}$$

Assume $d=4$ in (5), and $\tau=0.1$. We choose $ni=5674$ uniformly distributed interpolation points inside the domain and $nb1=nb2=150$. For $\alpha=0.35$, the sparsity of matrix Φ_t is 12.4%.

In this case, the discrete noisy data $\tilde{g}_t^{m+1}(\mathbf{x}_i)=g_t^{m+1}(\mathbf{x}_i)+\sigma*\mathrm{rand}(i)$ and $\tilde{q}_t^{m+1}(\mathbf{x}_i^*)=q_t^{m+1}(\mathbf{x}_i^*)+\sigma*\mathrm{rand}(i)$ are used, where $\sigma=0.1$. The relative error and RMSE of u at various t using the proposed method are listed in Table 4.

We can further observe the effectiveness and stability of this method for solving the IHCP from Table 4. Fig. 3 shows the approximate and exact temperature on $\partial\Omega_2$ at $t=1$.

6. Conclusion

We present a local meshless algorithm based on CS-RBFs. This algorithm is used to solve a certain kind of inverse problems for Poisson-type equations and an IHCP with certain sensors in the domain. The numerical results show that the proposed method can effectively solve the inverse problem with a large number of interpolation points. In solving the IHCP, we can see that this method can be easily extended to solving problems with irregular domains. Since this local method allows us to employ a lager amount of interpolation points if conditions permit, it is desirable

to apply the proposed method to more challenging 3D inverse problems. The implementation of the proposed method to 3D inverse problems is currently under investigation.

Acknowledgement

The first author acknowledges the support of Science and Technology Foundation Platform Construction Project of Shanxi Province of China (Project no. 2012091003-0101) and the Youth Fund of Taiyuan University of Technology (Project no. 2013T058, Project no. 2013T054). The second author thanks the Youth Fund of Taiyuan University of Technology (Project no. 2013T059) and Natural Science Foundation of Shanxi Province (Project no. 2013011002-3).

References

[1] Ling Leevan, Tomoya Takeuchi. Boundary control for inverse cauchy problems of the Laplace equations. Comput Model Eng Sci 2008;29:45–54.
[2] Li M, Xiong XT, Li Y. Method of fundamental solution for an inverse heat conduction problem with variable coefficients. Int J Comput Methods 2013;10.
[3] Lesnic D, Zeb A. The method of fundamental solutions for an inverse internal boundary value problem for the biharmonic equation. Int J Comput Methods 2009;6:557–67.
[4] Hon YC, Wei T. A fundamental solution method for inverse heat conduction problem. Eng Anal Bound Elem 2004;28:489–95.
[5] Wei T, Li YS. An inverse boundary problem for one-dimensional heat equation with a multilayer domain. Eng Anal Bound Elem 2009;33:225–32.
[6] Pourgholi R, Rostamian M. A stable numerical algorithm for solving an inverse parabolic problem. J Inf Comput Sci 2009;4:290–8.
[7] Marin L. An alternating iterative MFS algorithm for the Cauchy problem for the modified Helmholtz equation. Comput Mech 2010;45:665–77.
[8] Hu XY, Xu X, Chen WB. Numerical method for the inverse heat transfer problem in composite materials with Stefan–Boltzmann conditions. Adv Comput Math 2010;33:471–89.
[9] Liu GR, Lee JH, Patera AT, Yang ZL, Lam KY. Inverse identification of thermal parameters using reduced-basis method. Comput Methods Appl Mech Eng 2005;194:3090–107.
[10] Liu GR, Zaw K, Wang YY. Rapid inverse parameter estimation using reduced-basis approximation with asymptotic error estimation. Comput Methods Appl Mech Eng 2008;197:3898–910.
[11] Deng B, Tan KBC, Lu Y, Zaw K, Zhang J, Liu GR, et al. Inverse identification of elastic modulus of dental implant-bone interfacial tissue using neural network and FEA model. Inverse Probl Sci Eng 2009;17:1073–83.
[12] Fairweather G, Karageorghis A. The method of fundamental solutions for elliptic boundary value problems. Adv Comput Math 1998;9:69–95.
[13] Golberg MA, Chen CS. The theory of radial basis functions applied to the BEM for inhomogeneous partial differential equations. Bound Elem Commun 1994;5:57–61.
[14] Golberg MA, Chen CS, Bowman H, Power H. Some comments on the use of radial basis functions in the dual reciprocity method. Comput Mech 1998;21:141–8.
[15] Hon YC, Wu Z. A numerical computation for inverse boundary determination problem. Eng Anal Bound Elem 2000;24:599–606.
[16] Li J. A radial basis meshless method for solving inverse boundary value problems. Commun Numer Methods Eng 2004;20:51–60.
[17] Cheng AH-D, Cabral JJSP. Direct solution of ill-posed boundary value problems by radial basis function collocation method. Int J Numer Methods Eng 2005;64:45–64.
[18] Li M, Jiang TS, Hon YC. A meshless method based on RBFs method for nonhomogeneous backward heat conduction problem. Eng Anal Bound Elem 2010;34:785–92.
[19] Li M, Chen CS, Hon YC. A meshless method for solving nonhomogeneous Cauchy problems. Eng Anal Bound Elem 2011;35:499–506.
[20] Chen CS, Brebbia CA, Power H. Dual reciprocity method using compactly supported radial basis functions. Commun Numer Methods Eng 1999;15:137–50.
[21] Chen CS, Kuhn G, Li J, Mishuris G. Radial basis functions for solving near singular Poisson problems. Commun Numer Methods Eng 2003;19:333–47.
[22] Cheng AH-D, Young DL, Tsai CC. Solution of Poisson's equation by iterative DRBEM using compactly supported, positive definite radial basis function. Eng Anal Bound Elem 2000;24:549–57.
[23] Young DL, Tsai CC, Eldho TI, Cheng AH-D. Solution of Stokes Flow using an iterative DRBEM based on compactly supported, positive definite radial basis function. Int J Comput Math Appl 2002;43:607–19.
[24] Golberg MA, Chen CS, Ganesh M. Particular solution of 3D Helmholtz-type equations using compactly supported radial basis functions. Eng Anal Bound Elem 2000;24:539–47.
[25] Wong SM, Hon YC, Golberg MA. Compactly supported radial basis functions for shallow water equations. Appl Math Comput 2002;127:79–101.
[26] Wu ZM. Multivariate compactly supported positive definite radial functions. Adv Comput Math 2005;101:283–92.
[27] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Adv Comput Math 1995;4:389–96.
[28] Buhmann MD. A new class of radial basis functions with compact support. Math Comput 2001;70:307–18.
[29] Kansa EJ. Multiquadrics scattered data approximation scheme with applications to computational fluid dynamics I. Comput Math Appl 1990;19:127–45.
[30] Tikhonov AN, Arsenin VY. On the solution of ill-posed problems. New York: Wiley; 1977.
[31] Wei T, Hon YC, Ling L. Method of fundamental solutions with regularization techniques for Cauchy problems of elliptic operators. Eng Anal Bound Elem 2007;31:373–85.
[32] Lin J, Chen W, Wang FZ. A new investigation into regularization techniques for the method of fundamental solution. Math Comput Simul 2011;81:1144–52.
[33] Lawson CL, Hanson RJ. Solving least squares problems. Upper Saddle River, NJ: Prentice-Hall; 1974.
[34] Hansen PC, O'Leary DP. The use of the L-curve in the regularization of discrete ill-posed problems. SIAM J Sci Comput 1993;14:1487–503.
[35] Rashed YF. BEM for dynamic analysis using compact supported radial basis functions. Comput Struct 2002;80:1351–67.
[36] Hansen PC. Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems. Numer Algorithms 1994;6:1–35.