



ORIGINAL ARTICLE

# The MLPG with improved weight function for two-dimensional heat equation with non-local boundary condition

T. Techapirom <sup>c</sup>, A. Luadsong <sup>a,b,\*</sup>

<sup>a</sup> Department of Mathematics, Faculty of Science, King Mongkuts University of Technology Thonburi (KMUTT), 126 Pracha-utid Road, Bangmod, Toongkru, Bangkok 10140, Thailand

<sup>b</sup> Centre of Excellence in Mathematics, CHE, Si Ayutthaya, Bangkok 10400, Thailand

<sup>c</sup> Department of Mathematics and Applied Statistics, Faculty of Science and Technology, Bansomdejchaopraya Rajabhat University (BSRU), 1061 Soi Isaraphap Road, Dhonburi, Bangkok 10600, Thailand

Received 19 September 2012; accepted 22 February 2013

Available online 14 March 2013

## KEYWORDS

MLS approximation;  
MLPG method;  
Heat equation;  
Weight function;  
Non-local boundary condition;  
Kronecker delta function

**Abstract** In this paper, a meshless local Petrov–Galerkin (MLPG) method is presented to treat the heat equation with the Dirichlet, Neumann, and non-local boundary conditions on a square domain. The Moving Least Square (MLS) approximation is a classical MLS method, in which the Gaussian weight function is the most common shape function. However, shape functions for the classical MLS approximation lack the Kronecker delta function property. Thus in this method, the boundary conditions cannot have a penalty parameter imposed easily and directly. In the method we choose a weight function that leads to the MLS approximation shape functions approximating the Kronecker delta function property, and nodes on the Dirichlet boundary conditions, which enables a direct application of essential boundary conditions without the additional numerical method. The improved weight function in MLS approximation has been successfully implemented in solving the diffusion equation problem. Two test problems are presented to verify the efficiency, easiness and accuracy of the method. Also  $N_e$  and root mean square errors are obtained to show the convergence of the method.

© 2013 King Saud University. Production and hosting by Elsevier B.V. All rights reserved.

\* Corresponding author. Tel.: +66 24708837; fax: +66 24284025.  
E-mail addresses: anirut.lua@kmutt.ac.th, anirut.lua@gmail.com (A. Luadsong).

Peer review under responsibility of King Saud University.



\* The research is financed by: This research is partially supported by the Centre of Excellence in Mathematics, the Commission on Higher Education, Thailand.

## 1. Introduction

In the past several decades, there has been a growing interest in the development of meshless numerical techniques as alternatives to classical mesh-dependent numerical methods. The MLPG method was first discovered by Atluri and Zhu (1998). This method has been described in textbooks (Atluri, 2004; Liu and Gu, 2005) as allowing for freedom to choose the test function. The method is based on local weak forms and moving least square (MLS) approximation, and obtains

the true solution of the problem. The main advantage of the MLPG method is that it only requires nodes and a description of the external and internal boundary conditions, therefore, no element connectivity, neither total nor part, is needed. Effective implementations of meshless methods are a key to success (Katz, 2009; Dehghan, 2005; Liu, 2009; Atluri and Zhu, 1998, 2000; Atluri and Shengping, 2002; Abbasbandy and Shirzadi, 2011; Wu and Tao, 2008).

Finding the numerical solution with a non-local boundary condition is important for research in many fields of science and engineering such as chemical diffusion, diffusion equation, thermoelasticity, heat conduction process, heat transfer, control theory, medical schemes and so on (Martin-Vaquero and Vigo-Aguiar, 2009; Abbasbandy and Shirzadi, 2011; Kazem and Rad, 2012; Bogoya et al., 2012; Syed and Ahmet, 2011; Pisano et al., 2009). It is most widely used and very important in thermoelasticity. In 1963, Cannon (1963) first introduced non-local boundary condition problems, and most investigations developed various problems with one dimension, two dimension, the Dirichlet boundary condition, and the Neumann boundary condition. In 2010, Abbasbandy and Shirzadi (2010, 2011) researched on the MPLG method for the two-dimensional diffusion equation with the Neumann boundary condition and non-classical boundary condition, and a meshless method for the two-dimensional diffusion equation with an integral condition. The proposed method worked very well for the two-dimensional diffusion equations with a non-classical boundary condition, because of its simplicity and high accuracy.

The purpose of this paper is to improve the weight function in the MLS approximation to make the meshless method very efficient for solving the following two-dimensional time-dependent heat equation with non-local boundary conditions, given by Abbasbandy and Shirzadi (2011):

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}; \quad (x, y) \in \Omega \quad (1)$$

with initial condition

$$u(x, y, 0) = f(x, y), \quad \Omega = (x, y) \mid 0 \leq x, y \leq 1, \quad (2)$$

and boundary conditions

$$\frac{\partial u(0, y, t)}{\partial x} = g_0(y, t), \quad 0 \leq t \leq T, \quad 0 \leq y \leq 1, \quad (3)$$

$$\frac{\partial u(1, y, t)}{\partial x} = g_1(y, t), \quad 0 \leq t \leq T, \quad 0 \leq y \leq 1, \quad (4)$$

$$u(x, 1, t) = h_1(x, t), \quad 0 \leq t \leq T, \quad 0 \leq x \leq 1, \quad (5)$$

$$u(x, 0, t) = h_0(x)\mu(t), \quad 0 \leq t \leq T, \quad 0 \leq x \leq 1, \quad (6)$$

and the non-local boundary condition

$$\int_{\Omega} u(x, y, t) d\Omega = m(t), \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1, \quad (7)$$

where  $f$ ,  $g_0$ ,  $g_1$ ,  $h_0$ ,  $h_1$  and  $m$  are given functions, while the functions  $u$  and  $\mu$  are unknowns. The non-local boundary condition is variable-separable, with spatial dependence given by  $h_0(x)$  and time dependence given by  $\mu(t)$ .

Abbasbandy and Shirzadi (2011) used a meshless local Petrov–Galerkin (MLPG) method to treat parabolic partial differential equations with Dirichlet and Neumann conditions and a non-classical boundary. A difficulty in implementing the MLPG method to impose essential boundary conditions is that the

Moving Least Square (MLS) trial functions do not pass through the nodal values. To overcome this difficulty, they used the MLPG method only inside the domain, while at boundaries, they used finite difference schemes in all boundary conditions.

Most and Bucher (2005) presented the application of the MLS interpolation is solved with a new weighting function, which makes the MLS interpolation more attractive especially within a Galerkin method. A new weighting function was designed for meshless shape functions to fulfill these essential conditions with a very high accuracy without any additional effort. Due to the approximative character of this interpolation the obtained shape functions do not fulfill the interpolation conditions, which causes an additional numerical effort for the application of the boundary conditions. This will be clear by pointing out, that the choice of the base polynomial is arbitrary, thus the accuracy can be increased by choosing higher order polynomials.

Yang et al. (2011) used an improved hybrid boundary node method (hybrid BNM) for solving steady fluid flow problems, Miao et al. (2005) used a meshless hybrid boundary-node method for Helmholtz problems, and Wang et al. (2011) used the multi-domain hybrid boundary node method for 3D elasticity. This research used the Hybrid BNM, proposed by Zhang et al. (2002), for potential and elasticity problems which is a new boundary type meshless method and has been developed by Miao et al. (2005). Belytschko et al. (1996) combined MLS approximation and a modified variational principle. It only requires nodes constructed on the boundary of the domain, and does not require any mesh for the interpolation of variables, nor for the integration. The accuracy of the hybrid BNM is rather high. However, shape functions for the classical MLS approximation lack the delta function property, due to which Yang et al. (2011) in their paper used the improved MLS approach in the hybrid boundary node method. They proposed adopting a regularized weight function and used a new weight function that was designed by Most and Bucher. This method leading to the MLS shape functions fulfilling the interpolation condition exactly, enables a direct application of essential boundary conditions without additional numerical effort.

The objective of the present study is to improve the weight function in the classical MLS method. The Gaussian weight function is the most common; however, this kind of shape function does not have the Kronecker delta function property, so the boundary condition cannot be enforced easily and directly. Consequently, a transformation strategy of boundary condition application is inevitable, therefore, additional computational efforts cannot be avoided. In order to remove it, Most and Bucher designed a new weight function that allows for the fulfillment of the MLS interpolation condition with a very high accuracy.

In this paper, we implement the problems under a non-local boundary condition as in Eq. (7). Also, we used the weight function (Most and Bucher, 2005) in our implementation, for which the MLS approximation lacks the Kronecker delta function property, so it should be noted that the weight function leads to the MLS approximation shape functions fulfilling the interpolation condition exactly and enables a direct application of the Dirichlet boundary conditions, where solving the two-dimensional diffusion equation is shown to be accurate and less coding computational expense can be obtained.

The contents of this paper are as follows: Section 2 presents the improved MLS approach, Section 3 discusses the local weak

formulation of the problem, Section 4 describes the numerical implementation of the method for the diffusion equation with regular nodes, and finally Section 5 presents the conclusions.

## 2. The improved MLS approach

Consider a sub-domain  $\Omega_x$ , the neighborhood of a point  $\mathbf{x}$  and denoted as the domain of definition of the MLS approximation for the trial function at  $\mathbf{x}$ , which is located in the problem domain  $\Omega_x$ . To approximate the distribution of function  $u$  in  $\Omega_x$ , over a number of randomly located nodes  $\mathbf{x}_i$ ,  $i = 1, 2, \dots, n$ , the moving least square approximation  $u^h(\mathbf{x})$  of  $u$ ,  $\forall \mathbf{x} \in \Omega_x$  can be defined by

$$u^h(\mathbf{x}) = \mathbf{P}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \quad \forall \mathbf{x}, \quad (8)$$

where  $\mathbf{P}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x})]$  is a complete monomial basis of order  $m$ ; and  $\mathbf{a}(\mathbf{x})$  is a vector containing coefficients  $a_j(\mathbf{x})$ ,  $j = 1, 2, \dots, m$ , which are functions of the space coordinates  $\mathbf{x}$ . For example, for a 2-D problem,  $\mathbf{P}^T(\mathbf{x}) = [1, x, y]$  and  $\mathbf{P}^T(\mathbf{x}) = [1, x, y, x^2, xy, y^2]$ , for linear basis ( $m = 3$ ) and quadratic basis ( $m = 6$ ), respectively. The coefficient vector  $\mathbf{a}(\mathbf{x})$  is determined by minimizing a weighted discrete  $L_2$  norm, defined as

$$\begin{aligned} J(\mathbf{x}) &= \sum_{i=1}^n w_i(\mathbf{x}) [p^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - \hat{u}_i]^2 \\ &= [\mathbf{P} \cdot \mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}]^T \mathbf{W} [\mathbf{P} \cdot \mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}], \end{aligned} \quad (9)$$

where  $w_i(\mathbf{x})$  is the weight function associated with the node  $i$ , with  $w_i(\mathbf{x}) > 0$  for all  $\mathbf{x}$  in the support of  $w_i(\mathbf{x})$ ,  $\mathbf{x}_i$  denotes the value of  $\mathbf{x}$  at node  $i$ ,  $n$  is the number of nodes in  $\Omega_x$  for which the weight functions  $w_i(\mathbf{x}) > 0$ , the matrices  $\mathbf{P}$  and  $\mathbf{W}$  are defined as

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}^T(\mathbf{x}_1) \\ \mathbf{P}^T(\mathbf{x}_2) \\ \dots \\ \mathbf{P}^T(\mathbf{x}_n) \end{pmatrix}_{n \times m}, \quad \mathbf{W} = \begin{pmatrix} w_1(\mathbf{x}) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_n(\mathbf{x}) \end{pmatrix}$$

and  $\hat{\mathbf{u}}^T = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n]$ . Here it should be noted that  $\hat{u}_i$ ,  $i = 1, 2, \dots, n$ , in (9) are fictitious nodal values, and not the nodal values of the unknown trial function  $u^h(\mathbf{x})$  in general. The stationary point of  $J$  in (9) with respect to  $\mathbf{a}(\mathbf{x})$  leads to the following linear relation between  $\mathbf{a}(\mathbf{x})$  and  $\hat{\mathbf{u}}$

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\hat{\mathbf{u}}, \quad (10)$$

where the matrices  $\mathbf{A}(\mathbf{x})$  and  $\mathbf{B}(\mathbf{x})$  are defined by

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T \mathbf{W} \mathbf{P} = \sum_{i=1}^n w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}_i) \mathbf{p}^T(\mathbf{x}_i), \quad (11)$$

$$\begin{aligned} \mathbf{B}(\mathbf{x}) &= \mathbf{P}^T \mathbf{W} \\ &= [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}_1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}_2), \dots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}_n)]. \end{aligned} \quad (12)$$

The MLS approximation is well-defined only when the matrix  $\mathbf{A}$  in (10) is non-singular. It can be seen that this is the case if and only if the rank of  $\mathbf{P}$  equals  $m$ . A necessary condition for a well-defined MLS approximation is that at least  $m$  weight functions are non-zero (i.e.,  $n > m$ ) for each sample point  $\mathbf{x} \in \Omega$  and that the nodes in  $\Omega_x$  are not arranged in a special pattern, such as on a straight line. Here, a sample point may be a nodal point under consideration or a quadrature point.

Solving  $\mathbf{a}(\mathbf{x})$  from (10) and substituting it into (8) gives a relation which may be written in the form of an interpolation function similar to that used in FEM, as:

$$u^h(\mathbf{x}) = \Phi^T(\mathbf{x}) \cdot \hat{\mathbf{u}} = \sum_{i=1}^n \phi_i(\mathbf{x}) \hat{u}_i, \quad u^h(\mathbf{x}) \equiv u_i, \mathbf{x} \in \Omega_x \quad (13)$$

and essentially  $u_i \neq \hat{u}_i$ , where

$$\Phi^T(\mathbf{x}) = \mathbf{P}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x}), \quad (14)$$

or

$$\phi(\mathbf{x}) = \sum_{j=1}^m p_j(\mathbf{x}) [\mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})] \quad (15)$$

Usually  $\phi_i(\mathbf{x})$  is called the shape function of the MLS approximation corresponding to nodal point  $y_i$ . From (12) and (14), it may be seen that  $\phi_i(\mathbf{x}) = 0$  when  $w_i(\mathbf{x}) = 0$ . In practical applications,  $w_i(\mathbf{x})$  is generally chosen such that it is non-zero over the support of nodal points  $y_i$ . The support of the nodal point  $y_i$  is usually taken to be a circle of radius  $r_i$ , centered at  $y_i$ . The fact that  $\phi_i(\mathbf{x}) = 0$ , for  $\mathbf{x}$  not in the support of nodal point  $y_i$ , preserves the local character of the moving least square approximation.

Let  $C^q(\Omega)$  be the space of  $q^{\text{th}}$  continuously differentiable functions on  $\Omega$ . If  $w_i(\mathbf{x}) \in C^q(\Omega)$ ,  $p_j(\mathbf{x}) \in C^r(\Omega)$ ,  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, m$ , then  $\phi_i(\mathbf{x}) \in C^r(\Omega)$  with  $r = \min(q, s)$ .

The partial derivatives of  $\phi_i(\mathbf{x})$  are obtained as

$$\phi_{i,k}(\mathbf{x}) = \sum_{j=1}^m [p_{j,k}(\mathbf{A}^{-1}\mathbf{B})_{ji} + p_j(\mathbf{A}^{-1}\mathbf{B}_{,k} + \mathbf{A}_{,k}^{-1}\mathbf{B})_{ji}], \quad (16)$$

in which  $\mathbf{A}_{,k}^{-1} = (\mathbf{A}^{-1})_{,k}$  represents the derivative of the inverse of  $\mathbf{A}$  with respect to  $\mathbf{x}_k$ , which is given by  $\mathbf{A}_{,k}^{-1} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}$ , where  $(\bullet)_{,i}$  denotes  $\frac{\partial(\bullet)}{\partial x_i}$ .

In this paper, we should first choose the weight function in the implementation of the MLS approximation. In the classical MLS method, the Gaussian weight function is the most common. However, this kind of shape function does not have the Kronecker delta function property, so the boundary condition cannot be enforced easily and directly. Consequently, a transformation strategy of boundary condition application is inevitable, therefore, additional computational effort cannot be avoided. In order to remove it, Most and Bucher (2005) designed a new weight function which allows the fulfillment of the MLS interpolation condition with a very high accuracy:

$$\phi_i^{MLS}(\mathbf{x}_j) \approx \delta_{ij}. \quad (17)$$

The new weight function (Most and Bucher, 2005) is used as

$$w_R(\mathbf{x}) = \frac{\tilde{w}_R(\mathbf{x})}{\sum_{j=1}^k \tilde{w}_R(\mathbf{x}_j)}, \quad (18)$$

where

$$\tilde{w}_R(\mathbf{x}) = \begin{cases} \frac{(s^2 + \epsilon)^{-2} - (1 + \epsilon)^{-2}}{\epsilon^{-2} - (1 + \epsilon)^{-2}}, & 0 \leq s \leq 1 \\ 0 & s \geq 1. \end{cases} \quad (19)$$

The variable  $k$  belongs to the number of supporting points influencing  $\mathbf{x}$ ;  $s$  is the normalized distance between the interpolation point and the considered supporting point,  $s_i = \frac{\|\mathbf{x} - \mathbf{x}_i\|}{\mathbf{D}}$  and  $\mathbf{D}$  is the influence radius.

Most and Bucher (2005) recommended the regularization parameter  $\epsilon$ , which should be very small, as

$$\varepsilon = 10^{-5}. \quad (20)$$

### 3. Local weak formulation

The MLPG method constructs the weak form over local sub-domains such as  $\Omega_s$ , which is a small region taken for each node in the global domain  $\Omega$  and may be of any geometric shape and size. In this paper they are taken to be of circular shape. Because the weak form is constructed over local sub-domains, the formulation is called the ‘‘local weak formulation’’.

The local weak form of (1) for  $\mathbf{x}_i = (x^i, y^i) \in \Omega_s^i$  can be written as follows:

$$\int_{\Omega_s^i} \left( \frac{\partial u}{\partial t} - \nabla^2 u \right) v_i d\Omega = 0, \quad (21)$$

where  $v_i$  is a test function. Using the divergence theorem and  $(\nabla^2 u)v_i = \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} v_i \right) + \frac{\partial}{\partial y} \left( \frac{\partial u}{\partial y} v_i \right) - \left( \frac{\partial u}{\partial x} \frac{\partial v_i}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v_i}{\partial y} \right)$  yields the following expression:

$$\int_{\Omega_s^i} \frac{\partial u}{\partial t} v_i d\Omega - \int_{\partial\Omega_s^i} \frac{\partial u}{\partial n} v_i d\Gamma + \int_{\Omega_s^i} \frac{\partial u}{\partial x} \frac{\partial v_i}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v_i}{\partial y} d\Omega = 0, \quad (22)$$

where  $\Omega_s^i$  is a circle of radius  $r_0$  centered at  $\mathbf{x}_i$ ,  $\partial\Omega_s^i$  is the boundary of  $\Omega_s^i$ ,  $\mathbf{n} = (n_1, n_2)$  is the outward unit normal to the boundary  $\partial\Omega_s^i = \Gamma_{su}^i \cup \Gamma_{sq_0}^i \cup \Gamma_{sq_1}^i$ ,

$$\frac{\partial u}{\partial n} = \frac{\partial u}{\partial x} n_1 + \frac{\partial u}{\partial y} n_2,$$

and yields the following expression:

$$\int_{\Omega_s^i} \frac{\partial u}{\partial t} v_i d\Omega - \int_{\Gamma_{su}^i} \frac{\partial u}{\partial n} v_i d\Gamma - \int_{\Gamma_{sq_0}^i} \frac{\partial u}{\partial n} v_i d\Gamma - \int_{\Gamma_{sq_1}^i} \frac{\partial u}{\partial n} v_i d\Gamma + \int_{\Omega_s^i} \frac{\partial u}{\partial x} \frac{\partial v_i}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v_i}{\partial y} d\Omega = 0. \quad (23)$$

Substituting trial function  $u^h(\mathbf{x}, t) = \sum_{j=1}^n \phi_j(\mathbf{x}) \hat{u}_j(t)$  and  $q \equiv \frac{\partial u}{\partial n}$  into (23) yields:

$$\int_{\Omega_s^i} v_i \sum_{j=1}^n \phi_j \frac{\partial \hat{u}_j}{\partial t} d\Omega - \int_{\Gamma_{su}^i} v_i \sum_{j=1}^n \frac{\partial \phi_j}{\partial n} \hat{u}_j d\Gamma - \int_{\Gamma_{sq_0}^i} g_0 v_i d\Gamma - \int_{\Gamma_{sq_1}^i} g_1 v_i d\Gamma + \int_{\Omega_s^i} \sum_{j=1}^n \frac{\partial \phi_j}{\partial x} \frac{\partial v_i}{\partial x} \hat{u}_j + \sum_{j=1}^n \frac{\partial \phi_j}{\partial y} \frac{\partial v_i}{\partial y} \hat{u}_j d\Omega = 0, \quad (24)$$

$$\sum_{j=1}^n \int_{\Omega_s^i} v_i \phi_j d\Omega \frac{\partial \hat{u}_j}{\partial t} - \sum_{j=1}^n \int_{\Gamma_{su}^i} v_i \frac{\partial \phi_j}{\partial n} d\Gamma \hat{u}_j + \sum_{j=1}^n \int_{\Omega_s^i} \times \frac{\partial \phi_j}{\partial x} \frac{\partial v_i}{\partial x} + \frac{\partial \phi_j}{\partial y} \frac{\partial v_i}{\partial y} d\Omega \hat{u}_j = \int_{\Gamma_{sq_0}^i} g_0 v_i d\Gamma + \int_{\Gamma_{sq_1}^i} g_1 v_i d\Gamma, \quad (25)$$

where  $i = 1, 2, 3, \dots, n$ .

By substituting (13) into (23), the governing equations are transformed in the discretized system, written in a matrix form as

$$\mathbf{K} \frac{\partial \tilde{\mathbf{U}}}{\partial t} + \mathbf{F} \tilde{\mathbf{U}} = \mathbf{C}(t), \quad (26)$$

where,  $\mathbf{K}$ ,  $\mathbf{F}$  and  $\mathbf{C}$  are matrices described as follows:

$$\mathbf{K} = [K_{ij}], \quad K_{ij} = \int_{\Omega_s^i} \phi_j v_i d\Omega, \quad (27)$$

$$\begin{aligned} \mathbf{F} &= [F_{ij}], \quad F_{ij} \\ &= \int_{\Omega_s^i} \frac{\partial \phi_j}{\partial x} (v_i)_{,x} + \frac{\partial \phi_j}{\partial y} (v_i)_{,y} d\Omega + \int_{\Gamma_{sq_0}^i} v_i \frac{\partial \phi_j}{\partial y} d\Gamma - \int_{\Gamma_{su}^i} v_i \\ &\quad \times \frac{\partial \phi_j}{\partial y} d\Gamma, \end{aligned} \quad (28)$$

$$\mathbf{C}(t) = [C_i(t)], \quad C_i(t) = \int_{\Gamma_{sq_0}^i} g_0 v_i d\Gamma + \int_{\Gamma_{sq_1}^i} g_1 v_i d\Gamma. \quad (29)$$

Setting a time-stepping scheme to overcome the time derivative and applying the Crank–Nicolson technique of approximation to (26) yields:

$$\mathbf{K} \frac{\tilde{\mathbf{U}}^{k+1} - \tilde{\mathbf{U}}^k}{\Delta t} + \frac{\mathbf{F}}{2} (\tilde{\mathbf{U}}^{k+1} + \tilde{\mathbf{U}}^k) = \mathbf{C}^{k+\frac{1}{2}} \quad (30)$$

which can be rearranged into:

$$(\mathbf{2K} + \Delta t \mathbf{F}) \tilde{\mathbf{U}}^{k+1} = (\mathbf{2K} - \Delta t \mathbf{F}) \tilde{\mathbf{U}}^k + 2\Delta t \mathbf{C}^{k+\frac{1}{2}} \quad (31)$$

where  $\tilde{\mathbf{U}}$  is a matrix described as follows:

$$\tilde{\mathbf{U}} = \begin{pmatrix} \hat{\mathbf{U}} \\ \hat{\boldsymbol{\mu}} \end{pmatrix}. \quad (32)$$

From (31), assuming that  $\hat{u}_i^k$ , for  $i = 1, 2, \dots, N$  and  $\hat{\boldsymbol{\mu}}^k$  are known, our aim is to compute  $\hat{u}_i^{k+1}$ , for  $i = 1, 2, \dots, N$  and  $\hat{\boldsymbol{\mu}}^{k+1}$ . Now we have  $N + 1$  unknowns so we need one equation to compute these unknowns, which can be obtained from the non-local boundary condition from (7). Substituting trial function  $u^h(\mathbf{x}) = \sum_{j=1}^n \phi_j(\mathbf{x}) \hat{u}_j$ , yields:

$$\begin{aligned} \int_{\Omega} (u^h)^{k+1}(\mathbf{x}) d\Omega &= \int_{\Omega} \sum_{j=1}^n \phi_j(\mathbf{x}) \hat{u}_j^{k+1} d\Omega \\ &= \sum_{j=1}^n \left[ \int_{\Omega} \phi_j(\mathbf{x}) d\Omega \right] \hat{u}_j^{k+1} = m^{k+1}, \end{aligned} \quad (33)$$

which can be written in a matrix form as

$$\mathbf{S} \hat{\mathbf{U}}^{k+1} = m^{k+1}, \quad (34)$$

where  $\mathbf{S}$  is a matrix described as follows:

$$\mathbf{S} = [S_j], \quad S_j = \int_{\Omega} \phi_j(\mathbf{x}) d\Omega. \quad (35)$$

We use the matrix forms in (31) and (34) to compute  $\hat{u}_i^{k+1}$  and  $\hat{\boldsymbol{\mu}}^{k+1}$  for nodes located inside the domain and on the Neumann boundary conditions. For nodes on the Dirichlet boundary conditions, we have to impose boundary conditions as follows:

For nodes  $\mathbf{x}^l = (x^l, 1)$  on the top horizontal boundary ( $0 \leq x^l \leq 1$ ), using (5), we have

$$u_i^{k+1}(\mathbf{x}^l) = h_1(\mathbf{x}^l, (k+1)\Delta t). \quad (36)$$

For nodes  $\mathbf{x}^l = (x^l, 0)$  on the bottom horizontal boundary ( $0 \leq x^l \leq 1$ ), using (6), we have

$$u_i^{k+1} - h_0(\mathbf{x}^l) \hat{\boldsymbol{\mu}}^{k+1} = 0. \quad (37)$$

**Table 1** Values of RMS and  $N_e$  for Example 1 at time instant  $t = 0.5$  with using  $\Delta t = 0.1$ ,  $m = 3$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	7.3057e-1	7.2590e-2	1.0598e-1	1.1697e-1
25	3.5521e-1	3.7904e-2	5.5402e-2	1.0191e-1
36	2.4873e-1	2.6939e-2	3.7492e-2	8.2754e-2
81	1.1058e-1	1.2252e-2	1.2751e-2	4.2217e-2
121	7.4128e-2	8.2759e-3	7.4247e-3	3.0045e-2
289	3.8756e-2	4.3771e-3	2.7789e-3	1.7379e-2

**Table 2** Values of RMS and  $N_e$  for Example 1 at time instant  $t = 0.5$  with using  $\Delta t = 0.1$ ,  $m = 6$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	1.8850e-1	1.8729e-2	5.1431e-2	5.6761e-2
25	4.1526e-2	4.4312e-3	6.3143e-3	1.1615e-2
36	2.5892e-2	2.8043e-3	3.0335e-3	6.6957e-3
81	8.4987e-3	9.4160e-4	3.5117e-4	1.1627e-3
121	5.7840e-3	6.4575e-4	1.2698e-4	5.1384e-4
289	2.0057e-2	2.2652e-3	5.7904e-5	3.6213e-4

**Table 3** Values of RMS and  $N_e$  for Example 1 at time instant  $t = 1$  with using  $\Delta t = 0.1$ ,  $m = 3$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	2.1242e+0	7.7644e-2	3.3082e-1	1.3431e-1
25	8.9967e-1	3.5317e-2	1.3377e-1	9.0521e-2
36	6.2220e-1	2.4791e-2	8.5169e-2	6.9158e-2
81	2.6276e-1	1.0710e-2	2.5011e-2	3.0464e-2
121	1.7538e-1	7.2031e-3	1.3890e-2	2.0678e-2
289	9.3034e-2	3.8655e-3	5.0629e-3	1.1648e-2

**Table 4** Values of RMS and  $N_e$  for Example 1 at time instant  $t = 1$  with using  $\Delta t = 0.1$ ,  $m = 6$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	4.3870e-1	1.6036e-2	1.0908e-1	4.4288e-2
25	1.0064e-1	3.9508e-3	1.2502e-2	8.4599e-3
36	6.3443e-2	2.5278e-3	6.0435e-3	4.9074e-3
81	2.1955e-2	8.9485e-4	7.0239e-4	8.5552e-4
121	1.5056e-2	6.1836e-4	1.8767e-4	2.7938e-4
289	4.2948e-2	1.7844e-3	1.8851e-4	4.3371e-4

#### 4. Numerical experiments

For the improved MLS approximations, two test problems with regular nodes are presented to illustrate the efficiency and accuracy of the method. The new weight function is used for the MLS approximations, which the regularization parameter  $\varepsilon$  is chosen as  $\varepsilon = 10^{-5}$ . In this paper it is chosen as  $r_0 = 6h$ , where  $h$  is the distance between nodes in each direction and the analyzed domain is  $\Omega = [0,1] \times [0,1]$ . The errors

**Table 5** Values of RMS and  $N_e$  for Example 2 at time instant  $t = 0.5$  with using  $\Delta t = 0.1$ ,  $m = 3$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	1.3599e-1	2.7574e-2	3.4410e-2	6.2613e-2
25	3.5470e-2	7.5184e-3	1.5547e-3	4.7147e-3
36	2.5973e-2	5.5563e-3	3.4247e-4	1.2463e-3
81	1.3503e-2	2.9292e-3	8.4598e-4	4.6180e-3
121	9.6428e-3	2.1018e-3	6.3712e-4	4.2507e-3
289	6.98e-3	1.53e-3	4.46e-4	4.60e-3

**Table 6** Values of RMS and  $N_e$  for Example 2 at time instant  $t = 0.5$  with using  $\Delta t = 0.1$ ,  $m = 6$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	5.1141e-2	1.0370e-2	1.4069e-2	2.5599e-2
25	1.1087e-2	2.3501e-3	1.9632e-3	5.9538e-3
36	7.0657e-3	1.5115e-3	1.0018e-3	3.6456e-3
81	3.0529e-3	6.6229e-4	1.3558e-4	7.4007e-4
121	2.1725e-3	4.7352e-4	8.9797e-5	5.9911e-4
289	9.9018e-3	2.1736e-3	1.1148e-4	1.1495e-3

**Table 7** Values of RMS and  $N_e$  for Example 2 at time instant  $t = 1$  with using  $\Delta t = 0.1$ ,  $m = 3$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	1.8325e-1	2.2537e-2	3.4242e-2	3.7791e-2
25	6.0470e-2	7.7742e-3	1.4521e-3	2.6709e-3
36	4.2662e-2	5.5355e-3	1.8026e-3	3.9788e-3
81	1.9650e-2	2.5856e-3	8.3361e-4	2.7600e-3
121	1.3465e-2	1.7801e-3	5.0689e-4	2.0512e-3
289	1.01e-2	1.34e-3	3.83e-4	2.40e-3

**Table 8** Values of RMS and  $N_e$  for Example 2 at time instant  $t = 1$  with using  $\Delta t = 0.1$ ,  $m = 6$ .

$u(x,y,t)$			$\mu(t)$	
$n$	RMS	$N_e$	RMS	$N_e$
9	6.5064e-2	8.0019e-3	1.5036e-2	1.6595e-2
25	1.4472e-2	1.8606e-3	1.7848e-3	3.2829e-3
36	9.5892e-3	1.2442e-3	9.1070e-4	2.0102e-3
81	3.8241e-3	5.0318e-4	1.0290e-4	3.4071e-4
121	2.5400e-3	3.3580e-4	4.7439e-5	1.9197e-4
289	9.8013e-3	1.3050e-3	4.0534e-5	2.5350e-4

of  $u$  and  $\mu$  presented in the numerical results are presented by the root mean square (RMS) and  $N_e$  errors respectively where

$$\text{RMS} = \sqrt{\frac{\sum_{k=1}^N (u(\mathbf{x}_k) - \hat{u}(\mathbf{x}_k))^2}{N}}$$

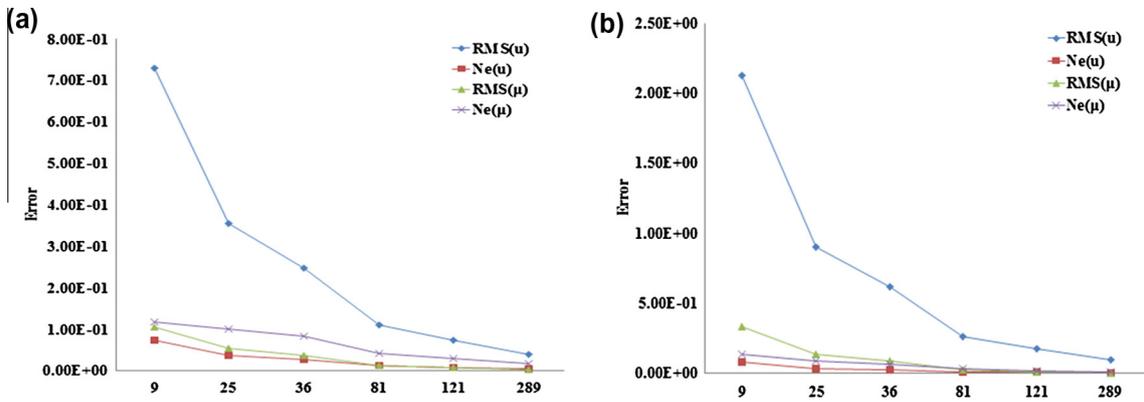


Figure 1 The error obtained at time instant  $t = 0.5$  (a),  $t = 1$  (b) and using  $\Delta t = 0.1$ ,  $m = 3$  with an increasing number of nodes for Example 1.

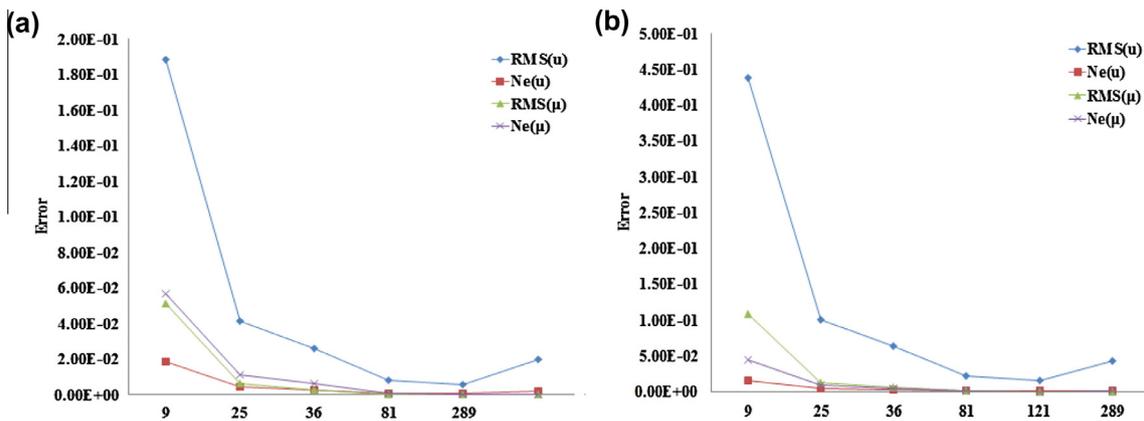


Figure 2 The error obtained at time instant  $t = 0.5$  (a),  $t = 1$  (b) and using  $\Delta t = 0.1$ ,  $m = 6$  with an increasing number of nodes for Example 1.

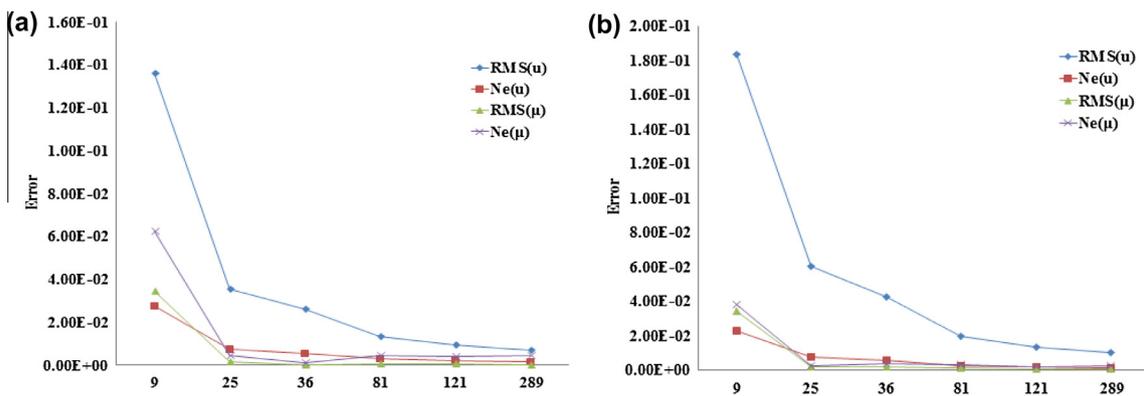
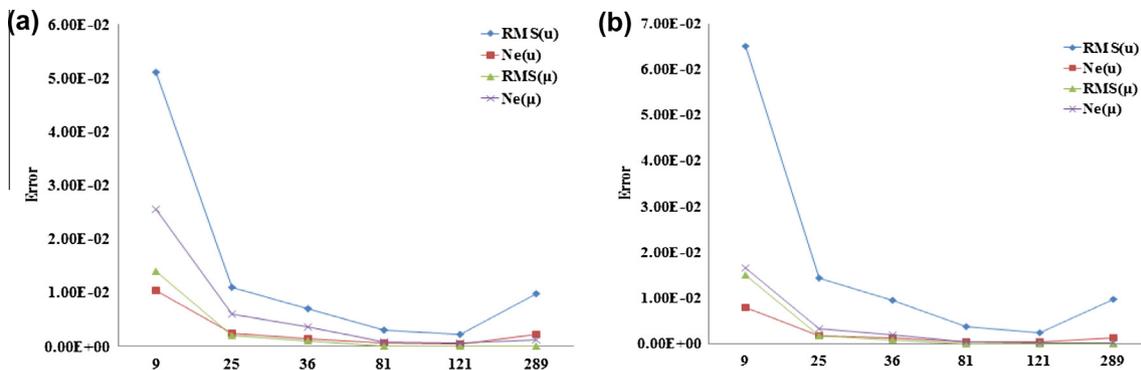


Figure 3 The error obtained at time instant  $t = 0.5$  (a),  $t = 1$  (b) and using  $\Delta t = 0.1$ ,  $m = 3$  with an increasing number of nodes for Example 2.

$$N_e = \sqrt{\frac{\sum_{k=1}^N (u(\mathbf{x}_k) - \hat{u}(\mathbf{x}_k))^2}{\sum_{k=1}^N u(\mathbf{x}_k)^2}}$$

$u(\mathbf{x}_k)$  and  $\hat{u}(\mathbf{x}_k)$  are achieved by an exact and approximate solution on  $\mathbf{x}_k$  points and  $N$  is the number of nodes.

The errors of the method for Example 1 at time instant  $t = 0.5$ ,  $\Delta t = 0.1$ ,  $m = 3$  are presented in Table 1 and  $m = 6$  is presented in Table 2. The errors of the method for Example 1 at time instant  $t = 1$ ,  $\Delta t = 0.1$ ,  $m = 3$  are presented in Table 3 and  $m = 6$  is presented in Table 4. The errors of the method for Example 2 at time instant  $t = 0.5$ ,  $\Delta t = 0.1$ ,  $m = 3$  are presented in Table 5 and  $m = 6$  is presented in Table 6. The errors of the method for Example 2 at time instant



**Figure 4** The error obtained at time instant  $t = 0.5$  (a),  $t = 1$  (b) and using  $\Delta t = 0.1$ ,  $m = 6$  with an increasing number of nodes for Example 2.

$t = 1$ ,  $\Delta t = 0.1$ ,  $m = 3$  are presented in Table 7 and  $m = 6$  is presented in Table 8. The numerical results in these tables show the convergence of the method by increasing the number of nodal points, which can see an increasing accuracy by increasing the number of nodal points. The error with an increasing number of nodes is presented in Figs. 1–4. The results reveal that the error decreases when the number of nodes increase.

In this paper, we referenced two examples from Abbasbandy and Shirzadi (2011) in a numerical scheme.

**Example 1.** For the first test problem with non-local boundary conditions, consider (1)–(7) with

$$\begin{aligned} f(x, y) &= \exp(x + y), g_0(y, t) = \exp(y + 2t), g_1(y, t) \\ &= \exp(1 + y + 2t), h_1(x, t) = \exp(1 + x + 2t), h_0(x) \\ &= \exp(x), m(t) = \exp(2t)(\exp(2) - 2\exp(1) + 1), \end{aligned}$$

for which the exact solution is  $u(x, y, t) = \exp(x + y + 2t)$ ,  $\mu(t) = \exp(2t)$ .

**Example 2.** For the second test problem with non-local boundary conditions, consider (1)–(7) with

$$\begin{aligned} f(x, y) &= (1 + y) \exp(x), g_0(y, t) = (1 + y) \exp(t), g_1(y, t) \\ &= (1 + y) \exp(1 + t), h_1(x, t) = 2 \exp(x + t), h_0(x) \\ &= \exp(x), \end{aligned}$$

$m(t) = \frac{3}{2}(\exp(1) - 1) \exp(t)$ , for which the exact solution is  $u(x, y, t) = (1 + y) \exp(x + t)$ ,  $\mu(t) = \exp(t)$ .

### 5. Conclusions

In this paper, an MLPG method was proposed for the study of two-dimensional heat equation with Dirichlet, Neumann, and non-local boundary conditions on a square domain. This method improved the classic MLS approximation by changing the Gaussian weight function, due to the shape functions constructed in the classical MLS that lack the delta function property so that the boundary condition cannot be enforced directly and a transformation strategy is indispensable. The efficiency has been greatly improved by using the improved MLS approach and the boundary conditions can be enforced

easily and directly imposing the Dirichlet boundary conditions and the coding work can also be reduced. Two numerical examples are presented to treat heat equation problems. The numerical results have demonstrated the accuracy, effectiveness of the present method, and the computation expense is decreased slightly since the boundary conditions can be enforced easily and directly imposing the Dirichlet boundary conditions. The values of  $N_e$  and RMS errors corroborate the appropriate accuracy of our method. It indicates that the method we introduced in this paper can easily be implemented to other problems.

### Acknowledgements

This research is partially supported by the Centre of Excellence in Mathematics, the Commission on Higher Education, Thailand. The authors would like to thank their adviser for providing advice and taking care of this research, and BSRU for providing a scholarship. Finally, the authors would like to thank their parents and family, who have also been encouraging, as well as their friends for taking care of each other.

### References

Abbasbandy, S., Shirzadi, A., 2010. A meshless method for two-dimensional diffusion equation with an integral condition. *Eng. Anal. Bound. Elem.* 34, 1031–1037.

Abbasbandy, S., Shirzadi, A., 2011. MLPG method for two-dimensional diffusion equation with Neumann’s and non-classical boundary conditions. *Appl. Numer. Math.* 61, 170–180.

Atluri, S.N., Shengping, Shen., 2002. The meshless local Petrov-Galerkin (MLPG) method: a simple and less-costly alternative to the finite element and boundary element methods. *CMES* 3, 11–51.

Atluri, S.N., Zhu, T.L., 1998. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput. Mech.* 22, 117–127.

Atluri, S.N., Zhu, T.L., 2000. The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics. *Comput. Mech.* 25, 169–179.

Atluri, S.N., 2004. *The Meshless Method (MLPG) for Domain and Bie Discretizations*. Tech Science Press, GA, USA.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P., 1996. Meshless method: an overview and recent development. *Comput. Method Appl. Mech. Eng.* 139, 3–47.

- Bogoya, M., Cesar, A., Gomez, S., 2012. On a nonlocal diffusion model with Neumann boundary conditions. *Nonlinear Anal.* 75, 3198–3209.
- Cannon, J.R., 1963. The solution of the heat equation subject to the specification of energy. *Quart. Appl. Math.* 21, 155–160.
- Dehghan, M., 2005. The one-dimensional heat equation subject to a boundary integral specification. *Appl. Math.* 32, 661–675.
- Katz, A.J., 2009. *Meshless Methods for Computational Fluid Dynamics*, Ph.D Thesis, Stanford University.
- Kazem, S., Rad, J.A., 2012. Radial basis functions method for solving of a non-local boundary value problem with Neumann's boundary conditions. *Appl. Math. Model.* 36, 2360–2369.
- Liu, G.R., Gu, Y.T., 2005. *An Introduction to Meshfree Methods and their Programming*. Springer, The Netherlands.
- Liu, G.R., 2009. *Meshfree Methods: Moving Beyond the Finite Element Method*/G.R.Liu, second ed. CRC Press, USA.
- Martin-Vaquero, J., Vigo-Aguiar, J., 2009. A note on efficient techniques for the second-order parabolic equation subject to non-local conditions. *Appl. Math.* 59, 1258–1264.
- Miao, Y., Wang, Y.H., Yu, F., 2005. Development of hybrid boundary node method in two-dimensional elasticity. *Eng. Anal. Bound. Elem.* 29, 703–712.
- Most, T., Bucher, C., 2005. A moving least squares weighting function for the element-free Galerkin method which almost fulfills essential boundary conditions. *Struct. Eng. Mech.* 21, 315–332.
- Pisano, A.A., Sofi, A., Fuschi, P., 2009. Nonlocal integral elasticity: 2D finite element based solutions. *Int. J. Solids Struct.* 46, 3836–3849.
- Syed, T.M.D., Ahmet, Y., 2011. On two-dimensional diffusion with integral condition. *J. King Saud Univ. Sci.* 23, 121–125.
- Wang, Q., Zheng, J.J., Miao, Y., Lv, J.H., 2011. The multi-domain hybrid boundary node method for 3D elasticity. *Eng. Anal. Bound. Elem.* 35, 803–810.
- Wu, X.H., Tao, W.Q., 2008. Meshless method based on the local weak-forms for steady-state heat conduction problems. *Int. J. Heat Mass Transfer* 51, 3103–3112.
- Yang, Q.N., Zheng, J.J., Miao, Y., Sima, Y.Z., 2011. An improved hybrid boundary node method for solving steady fluid flow problems. *Eng. Anal. Bound. Elem.* 35, 18–24.
- Zhang, J.M., Yao, Z.H., Li, H., 2002. A hybrid boundary node method. *Int. J. Numer. Method Eng.* 53, 751–763.