Vol 18 No 10, October 2009
1674-1056/2009/18(10)/4065–09

Chinese Physics B

© 2009 Chin. Phys. Soc.
and IOP Publishing Ltd

An improved boundary element-free method (IBEFM) for two-dimensional potential problems*

Ren Hong-Ping(任红萍)[a], Cheng Yu-Min(程玉民)[b]†, and Zhang Wu(张 武)[a]

[a] *School of Computer Engineering and Science, Shanghai University, Shanghai* 200072, *China*

[b] *Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai* 200072, *China*

(Received 7 October 2008; revised manuscript received 1 February 2009)

The interpolating moving least-squares (IMLS) method is discussed first in this paper. And the formulae of the IMLS method obtained by Lancaster are revised. Then on the basis of the boundary element-free method (BEFM), combining the boundary integral equation (BIE) method with the IMLS method, the improved boundary element-free method (IBEFM) for two-dimensional potential problems is presented, and the corresponding formulae of the IBEFM are obtained. In the BEFM, boundary conditions are applied directly, but the shape function in the MLS does not satisfy the property of the Kronecker δ function. This is a problem of the BEFM, and must be solved theoretically. In the IMLS method, when the shape function satisfies the property of the Kronecker δ function, then the boundary conditions, in the meshless method based on the IMLS method, can be applied directly. Then the IBEFM, based on the IMLS method, is a direct meshless boundary integral equation method in which the basic unknown quantity is the real solution of the nodal variables, and the boundary conditions can be applied directly and easily, thus it gives a greater computational precision. Some numerical examples are presented to demonstrate the method.

Keywords: moving least-squares approximation, interpolating moving least-squares method, meshless method, improved boundary element-free method, potential problem

PACC: 0260, 0270, 1240Q

1. Introduction

In recent years, more and more attention has been paid to research on the meshless (or meshfree) method.[1−8] The meshless method has some advantages over the traditional computational methods, such as the finite element method (FEM) and the boundary element method (BEM).

The meshless boundary integral equation method is an important direction of the meshless method. Combining the different approximation function in meshless method with the boundary integral equation method, some meshless boundary integral equation methods have been developed, such as the boundary node method (BNM),[9−12] the local boundary integral equation (LBIE) method,[13−16] and the boundary element-free method (BEFM).[17−25]

The moving least-squares (MLS) approximation is one of the bases of the meshless method.[1,2,26] The meshless method based on the MLS approximation can obtain a solution that has great precision. Some important meshless boundary integral equation methods are formed by combining the MLS approximation with boundary integral equation methods.[9−25] The boundary node method (BNM) is one of the meshless boundary integral equation methods, and Mukherjee *et al*[9−12] presented it to solve potential problems and linear elasticity problems. Another important method of the meshless boundary integral equation methods is the local boundary integral equation (LBIE) method, which Atluri *et al*[13−16] presented to solve linear and nonlinear boundary value problems. In the BNM, the basic unknown quantities are approximations of the nodal variables. However, they are not the real nodal variables, and thus the boundary conditions cannot be directly applied. The LBIE method can easily solve a problem with a complicated boundary, but the local foundational solution is more complicated than it is when the conventional boundary element method is used. Moreover, in the LBIE method, the basic unknown quantities are also approximations of the nodal variables, and again, the boundary conditions cannot be applied directly. The BNM and the LBIE methods are the indirect meshless boundary integral equation methods.

*Project supported by the National Natural Science Foundation of China (Grant No 10871124), Innovation Program of Shanghai Municipal Education Commission (Grant No 09ZZ99) and Shanghai Leading Academic Discipline Project (Grant No J50103).

†Corresponding author. E-mail: ymcheng@shu.edu.cn, ymcheng@sh163.net

The MLS approximation was developed from the conventional least-squares method, and in practical numerical processes, the MLS approximation is the conventional least-squares method for every selected point. A disadvantage of the conventional least-squares method is that the final algebra equations system is sometimes ill-conditioned or singular. Then the MLS approximation was improved by Liew and Cheng, and the improved moving least-squares approximation was presented.[17] In the improved moving least-squares approximation, the algebra equation system is not ill-conditioned or singular, and the solution of the algebra equation system can be obtained directly without obtaining the inverse matrix. There are also fewer coefficients in the improved moving least-squares approximation than there are in the MLS approximation, and hence the computing speed and efficiency are greater. And combining the boundary integral equation method and the improved moving least-squares approximation, a direct meshless method of boundary integral equation, which is called boundary element-free method (BEFM), was presented to solve the problems, such as; elasticity, elastodynamics, and fracture.[18−25] The improved element-free Galerkin method based on the improved moving least-squares approximation was discussed by Zhang, Liew and Cheng.[27,28]

In the BEFM, the basic unknown quantities are the real solutions of the nodal variables, but in the BNM and the LBIE method, the basic unknown quantities are approximations of the nodal variables. The BEFM is a direct meshless method of the boundary integral equation. And the BEFM has a greater computational precision. In the BEFM, boundary conditions are applied directly, but the shape function in the improved moving least-squares approximation does not satisfy the property of the Kronecker δ function. Then we think this is a problem in the BEFM, and must be solved theoretically. The interpolating moving least-squares (IMLS) method presented by Lancaster[26] can obtain the shape function which satisfies the property of the Kronecker δ function. If we apply the IMLS to the BEFM, the boundary conditions can be applied directly without doubt.

In this paper, the IMLS method is discussed first. And the formulae of the IMLS obtained by Lancaster are revised. Then on the basis of the boundary element-free method (BEFM), combining the boundary integral equation method with the IMLS method, the improved boundary element-free method (IBEFM) for two-dimensional potential problems is presented, and the corresponding formulae of the IBEFM are obtained. The IBEFM is a direct meshless boundary integral equation method in which the basic unknown quantity is the real solution of the nodal variables, and the boundary conditions can be applied directly and easily without doubt, thus it gives a greater computational precision. Some numerical examples are presented to demonstrate the efficiency of the method.

2. Moving least-squares approximation

In the MLS approximation, the trial function is

$$u^h(\boldsymbol{x}) = \sum_{i=1}^{m} p_i(\boldsymbol{x}) a_i(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}), \tag{1}$$

where $p_i(\boldsymbol{x})$, i=1, 2, ..., m, are monomial basis functions, m is the number of terms in the basis, and $a_i(\boldsymbol{x})$ are the coefficients of the basis functions.

In general, the basis functions are as follows.

(i) Linear basis:

$$\boldsymbol{p}^{\mathrm{T}} = (1, x_1) \ \text{ in 1D}, \tag{2}$$

$$\boldsymbol{p}^{\mathrm{T}} = (1, x_1, x_2) \ \text{ in 2D}. \tag{3}$$

(ii) Quadratic basis:

$$\boldsymbol{p}^{\mathrm{T}} = (1, x_1, x_1^2) \ \text{ in 1D}, \tag{4}$$

$$\boldsymbol{p}^{\mathrm{T}} = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2) \ \text{ in 2D}. \tag{5}$$

The local approximation that is defined by Lancaster and Salkauskas[26] is

$$u^h(\boldsymbol{x}, \bar{\boldsymbol{x}}) = \sum_{i=1}^{m} p_i(\bar{\boldsymbol{x}}) a_i(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\bar{\boldsymbol{x}})\,\boldsymbol{a}(\boldsymbol{x}). \tag{6}$$

To precisely obtain the local approximation of the function $u(\boldsymbol{x})$, the difference between the local approximation $u^h(\boldsymbol{x})$ and the function $u(\boldsymbol{x})$ must be minimized by a weighted least-squares method.

Define a functional

$$\begin{aligned} J &= \sum_{I=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_I)[u^h(\boldsymbol{x}, \boldsymbol{x}_I) - u(\boldsymbol{x}_I)]^2 \\ &= \sum_{I=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_I) \left[\sum_{i=1}^{m} p_i(\boldsymbol{x}_I) \cdot a_i(\boldsymbol{x}) - u(\boldsymbol{x}_I)\right]^2, \end{aligned} \tag{7}$$

where $w(\boldsymbol{x} - \boldsymbol{x}_I)$ is a weight function with compact support, and $\boldsymbol{x}_I$, $I = 1, 2, \ldots, n$, are the nodes in a compact support domain of point $\boldsymbol{x}$.

Equation (7) can be written as

$$J = (\boldsymbol{P}\boldsymbol{a} - \boldsymbol{u})^{\mathrm{T}} \boldsymbol{W}(\boldsymbol{x})(\boldsymbol{P}\boldsymbol{a} - \boldsymbol{u}), \tag{8}$$

where

$$\boldsymbol{u} = (u_1, u_2, \ldots, u_n)^{\mathrm{T}} = (u(\boldsymbol{x}_1), u(\boldsymbol{x}_2), \ldots, u(\boldsymbol{x}_n))^{\mathrm{T}}, \tag{9}$$

$$\begin{aligned} \boldsymbol{P} &= \begin{bmatrix} p_1(\boldsymbol{x}_1) & p_2(\boldsymbol{x}_1) & \cdots & p_m(\boldsymbol{x}_1) \\ p_1(\boldsymbol{x}_2) & p_2(\boldsymbol{x}_2) & \cdots & p_m(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\boldsymbol{x}_n) & p_2(\boldsymbol{x}_n) & \cdots & p_m(\boldsymbol{x}_n) \end{bmatrix} \\ &= (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m), \end{aligned} \tag{10}$$

$$\boldsymbol{W}(\boldsymbol{x}) = \begin{bmatrix} w(\boldsymbol{x} - \boldsymbol{x}_1) & 0 & \cdots & 0 \\ 0 & w(\boldsymbol{x} - \boldsymbol{x}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(\boldsymbol{x} - \boldsymbol{x}_n) \end{bmatrix}, \tag{11}$$

$$\boldsymbol{a} = \boldsymbol{a}(\boldsymbol{x}) = (a_1(\boldsymbol{x}), a_2(\boldsymbol{x}), \ldots, a_n(\boldsymbol{x}))^{\mathrm{T}}. \tag{12}$$

On the space $\mathrm{span}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$, the following inner product

$$(f, g)_{\boldsymbol{x}} = \sum_{I=1}^{n} w(\boldsymbol{x} - \boldsymbol{x}_I) f(\boldsymbol{x}_I) g(\boldsymbol{x}_I) \tag{13}$$

is defined for $\forall f(\boldsymbol{x})$, $g(\boldsymbol{x}) \in \mathrm{span}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$. And the norm of $f(\boldsymbol{x})$ is defined as

$$\|f\|_{\boldsymbol{x}} = [(f, f)_{\boldsymbol{x}}]^{1/2}. \tag{14}$$

Here, a function with the subscript $\boldsymbol{x}$ represents the function that is related to the selected point $\boldsymbol{x}$ in $\boldsymbol{a}(\boldsymbol{x})$ and the weight function $w(\boldsymbol{x} - \boldsymbol{x}_I)$.

For the set of points $\{\boldsymbol{x}_I\}_{i=1}^n$ and weight functions $\{w_i\}$, if the functions $\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \ldots, \varphi_m(\boldsymbol{x})$ satisfy the following conditions

$$\begin{aligned} (\varphi_k, \varphi_j) &= \sum_{i=1}^{n} \varphi_k(\boldsymbol{x}_i) w_i \varphi_j(\boldsymbol{x}_i) \\ &= \begin{cases} 0 & k \neq j \\ A_k & k = j \end{cases} \quad (k, j = 1, 2, \cdots, m), \end{aligned} \tag{15}$$

then the weight function set $\varphi_1(\boldsymbol{x}), \varphi_2(\boldsymbol{x}), \ldots, \varphi_m(\boldsymbol{x})$ is called an orthogonal function set with the weight function $\{w_i\}$ about points $\{\boldsymbol{x}_i\}_{i=1}^n$.

From Eq.(8) we have

$$\begin{aligned} J &= \|\boldsymbol{P}\boldsymbol{a} - \boldsymbol{u}\|_{\boldsymbol{x}}^2 \\ &= \|a_1\boldsymbol{p}_1 + a_2\boldsymbol{p}_2 + \cdots + a_m\boldsymbol{p}_m - \boldsymbol{u}\|_{\boldsymbol{x}}^2, \end{aligned} \tag{16}$$

then

$$J = \left\|\boldsymbol{u}^{(1)} - \boldsymbol{u}\right\|_{\boldsymbol{x}}^2 \tag{17}$$

is the minimum of the functional J, where $\boldsymbol{u}^{(1)}$ is the projection of $\boldsymbol{u}$ in $\mathrm{span}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m)$.

Let

$$\boldsymbol{u} = \boldsymbol{u}^{(1)} + \boldsymbol{u}^{(2)}, \tag{18}$$

where

$$\boldsymbol{u}^{(1)} \in \mathrm{span}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m), \tag{19}$$

$$\boldsymbol{u}^{(2)} \perp \mathrm{span}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_m), \tag{20}$$

then

$$(\boldsymbol{p}_i, \boldsymbol{u}^{(2)})_{\boldsymbol{x}} = 0, \quad (i = 1, 2, \ldots, m), \tag{21}$$

and then we have

$$\begin{aligned} (\boldsymbol{p}_i, \boldsymbol{u})_{\boldsymbol{x}} = (\boldsymbol{p}_i, \boldsymbol{u}^{(1)})_{\boldsymbol{x}} + (\boldsymbol{p}_i, \boldsymbol{u}^{(2)})_{\boldsymbol{x}} \quad &= (\boldsymbol{p}_i, \boldsymbol{u}^{(1)})_{\boldsymbol{x}}, \\ (i = 1, 2, \ldots, m). \end{aligned} \tag{22}$$

From Eq.(19), we have

$$\begin{aligned} (\boldsymbol{p}_i, \boldsymbol{u})_{\boldsymbol{x}} = (\boldsymbol{p}_i, \boldsymbol{u}^{(1)})_{\boldsymbol{x}} = \left(\boldsymbol{p}_i, \sum_{j=1}^{m} a_j(\boldsymbol{x})\boldsymbol{p}_j\right)_{\boldsymbol{x}}, \\ (i = 1, 2, \ldots, m). \end{aligned} \tag{23}$$

Equation (23) can be written as

$$\begin{bmatrix} (\boldsymbol{p}_1, \boldsymbol{p}_1)_{\boldsymbol{x}} & (\boldsymbol{p}_1, \boldsymbol{p}_2)_{\boldsymbol{x}} & \cdots & (\boldsymbol{p}_1, \boldsymbol{p}_m)_{\boldsymbol{x}} \\ (\boldsymbol{p}_2, \boldsymbol{p}_1)_{\boldsymbol{x}} & (\boldsymbol{p}_2, \boldsymbol{p}_2)_{\boldsymbol{x}} & \cdots & (\boldsymbol{p}_2, \boldsymbol{p}_m)_{\boldsymbol{x}} \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{p}_m, \boldsymbol{p}_1)_{\boldsymbol{x}} & (\boldsymbol{p}_m, \boldsymbol{p}_2)_{\boldsymbol{x}} & \cdots & (\boldsymbol{p}_m, \boldsymbol{p}_m)_{\boldsymbol{x}} \end{bmatrix} \begin{bmatrix} a_1(\boldsymbol{x}) \\ a_2(\boldsymbol{x}) \\ \vdots \\ a_m(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} (\boldsymbol{p}_1, \boldsymbol{u})_{\boldsymbol{x}} \\ (\boldsymbol{p}_2, \boldsymbol{u})_{\boldsymbol{x}} \\ \vdots \\ (\boldsymbol{p}_m, \boldsymbol{u})_{\boldsymbol{x}} \end{bmatrix}, \tag{24}$$

i.e.

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}, \tag{25}$$

where matrices $\boldsymbol{A}(\boldsymbol{x})$ and $\boldsymbol{B}(\boldsymbol{x})$ are

$$\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{P}, \tag{26}$$

$$\boldsymbol{B}(\boldsymbol{x}) = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x}). \tag{27}$$

From Eq.(25) we can obtain

$$\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}. \tag{28}$$

The expression of the local approximation $u^h(\boldsymbol{x})$ is then

$$u^h(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{u} = \sum_{I=1}^{n}\phi_I(\boldsymbol{x})u_I, \tag{29}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is called the shape function and

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{x}) &= (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \cdots, \phi_n(\boldsymbol{x})) \\ &= \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{A}^{-1}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x}).\end{aligned} \tag{30}$$

When $m = 1$, the approximation function is

$$u^h(\boldsymbol{x}) = \frac{\sum\limits_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)u_I}{\sum\limits_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)}. \tag{31}$$

3. Interpolating moving least-squares method

On the space $\mathrm{span}(p_1, p_2, \ldots, p_m)$, normalizing $p_1(\boldsymbol{x})$ we have

$$\beta_{\boldsymbol{x}}^{(1)} = \frac{p_1}{\|p_1\|_{\boldsymbol{x}}} = \frac{1}{\left[\sum\limits_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)\right]^{1/2}}. \tag{32}$$

Then we can generate functions orthogonal to $\beta_{\boldsymbol{x}}^{(1)}(\boldsymbol{x})$ as follows:

$$\begin{aligned}b_{\boldsymbol{x}}^{(i)}(\boldsymbol{x}) &= p_i(\boldsymbol{x}) - (p_i, \beta_{\boldsymbol{x}}^{(1)})_{\boldsymbol{x}}\beta_{\boldsymbol{x}}^{(1)} \\ &= p_i(\boldsymbol{x}) - \frac{\sum\limits_{I=1}^{n} p_i(\boldsymbol{x}_I)w(\boldsymbol{x}-\boldsymbol{x}_I)}{\sum\limits_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)}, \\ &\quad (i = 2, 3, \ldots, m)\end{aligned} \tag{33a}$$

i.e.

$$b_{\boldsymbol{x}}^{(i)}(\boldsymbol{x}) = p_i(\boldsymbol{x}) - \sum_{I=1}^{n} p_i(\boldsymbol{x}_I)v(\boldsymbol{x}-\boldsymbol{x}_I), \tag{33b}$$

where

$$v(\boldsymbol{x}-\boldsymbol{x}_I) = \frac{w(\boldsymbol{x}-\boldsymbol{x}_I)}{\sum\limits_{I=1}^{n} w(\boldsymbol{x}-\boldsymbol{x}_I)}. \tag{34}$$

In the MLS, when the new basis functions $\beta_{\boldsymbol{x}}^{(1)}(\boldsymbol{x}), b_{\boldsymbol{x}}^{(2)}(\boldsymbol{x}), b_{\boldsymbol{x}}^{(3)}(\boldsymbol{x}), \ldots, b_{\boldsymbol{x}}^{(m)}(\boldsymbol{x})$ are used, the corresponding approximation function is

$$u^h(\boldsymbol{x}) = (u, \beta_{\boldsymbol{x}}^{(1)}(\boldsymbol{x}))_{\boldsymbol{x}}\beta_{\boldsymbol{x}}^{(1)}(\boldsymbol{x}) + \sum_{i=2}^{m} a_{i-1}(\boldsymbol{x})b_{\boldsymbol{x}}^{(i)}(\boldsymbol{x}), \tag{35}$$

i.e.

$$\boldsymbol{u}^h(\boldsymbol{x}) = \boldsymbol{v}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{b}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{a}(\boldsymbol{x}), \tag{36}$$

where

$$\boldsymbol{v}(\boldsymbol{x}) = (v(\boldsymbol{x}-\boldsymbol{x}_1), v(\boldsymbol{x}-\boldsymbol{x}_2), \ldots, v(\boldsymbol{x}-\boldsymbol{x}_n))^{\mathrm{T}}, \tag{37}$$

$$\boldsymbol{b}(\boldsymbol{x}) = (b_{\boldsymbol{x}}^{(2)}(\boldsymbol{x}), b_{\boldsymbol{x}}^{(3)}(\boldsymbol{x}), \ldots, b_{\boldsymbol{x}}^{(m)}(\boldsymbol{x}))^{\mathrm{T}}, \tag{38}$$

$$\boldsymbol{a}(\boldsymbol{x}) = (a_1(\boldsymbol{x}), a_2(\boldsymbol{x}), \ldots, a_{m-1}(\boldsymbol{x}))^{\mathrm{T}}. \tag{39}$$

Then using the MLS approximation, we have

$$\boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{A}_{\boldsymbol{x}}^{-1}(\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x}}(\boldsymbol{x})\boldsymbol{u}. \tag{40}$$

where

$$\boldsymbol{A}_{\boldsymbol{x}}(\boldsymbol{x}) = \boldsymbol{C}_{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{C}_{\boldsymbol{x}}, \tag{41}$$

$$\boldsymbol{B}_{\boldsymbol{x}}(\boldsymbol{x}) = \boldsymbol{C}_{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{x}), \tag{42}$$

$$\boldsymbol{C}_{\boldsymbol{x}} = \begin{bmatrix} b_{\boldsymbol{x}}^{(2)}(\boldsymbol{x}_1) & b_{\boldsymbol{x}}^{(3)}(\boldsymbol{x}_1) & \cdots & b_{\boldsymbol{x}}^{(m)}(\boldsymbol{x}_1) \\ b_{\boldsymbol{x}}^{(2)}(\boldsymbol{x}_2) & b_{\boldsymbol{x}}^{(3)}(\boldsymbol{x}_2) & \cdots & b_{\boldsymbol{x}}^{(m)}(\boldsymbol{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ b_{\boldsymbol{x}}^{(2)}(\boldsymbol{x}_n) & b_{\boldsymbol{x}}^{(3)}(\boldsymbol{x}_n) & \cdots & b_{\boldsymbol{x}}^{(m)}(\boldsymbol{x}_n) \end{bmatrix}. \tag{43}$$

Substituting Eq.(40) into Eq.(36), we have

$$u^h(\boldsymbol{x}) = \boldsymbol{v}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{b}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{A}_{\boldsymbol{x}}^{-1}(\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x}}(\boldsymbol{x})\boldsymbol{u}. \tag{44}$$

Equation (44) can be written as

$$u^h(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{u}, \tag{45}$$

where

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{v}^{\mathrm{T}}(\boldsymbol{x}) + \boldsymbol{b}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{A}_{\boldsymbol{x}}^{-1}(\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x}}(\boldsymbol{x}). \tag{46}$$

When the weight function $w(\boldsymbol{x}-\boldsymbol{x}_I)$ is defined as

$$w(\boldsymbol{x}-\boldsymbol{x}_I) = |\boldsymbol{x}-\boldsymbol{x}_I|^{-\alpha}, \tag{47}$$

where α is an even positive integer.

The function $u^h(\boldsymbol{x})$ in Eq.(45) can interpolate at all points $\{\boldsymbol{x}_I\}_{I=1}^{n}$, i.e.

$$u^h(\boldsymbol{x}_I) = u(\boldsymbol{x}_I). \tag{48}$$

Then we have

$$\phi_I(\boldsymbol{x}_J) = \begin{cases} 1, \ I = J \\ 0, \ I \neq J \end{cases}. \tag{49}$$

Corresponding to Eq.(44), reference [26] obtained the following equation

$$\boldsymbol{C}_{\boldsymbol{x}}^{\mathrm{T}} \boldsymbol{W}(\boldsymbol{x}) \boldsymbol{C}_{\boldsymbol{x}} \boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{C}_{\boldsymbol{x}}^{\mathrm{T}} \boldsymbol{W}(\boldsymbol{x})(\boldsymbol{u} - (\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{u})\boldsymbol{p}_1), \tag{50}$$

where $p_1(\boldsymbol{x}_I) = 1$.

Equation (44) is different from Eq.(50), then the revised formulae of the IMLS method are given in this paper.

4. Improved boundary element-free method for two-dimensional potential problems

In this Section, we study the boundary element-free method of the following two-dimensional Poisson's equation

$$\nabla^2 u(\boldsymbol{x}) = b(\boldsymbol{x}), \quad \text{in} \quad \Omega \tag{51}$$

with boundary conditions of the Dirichlet type, i.e.

$$u(\boldsymbol{x}) = \bar{u}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma_u \tag{52}$$

or the Neumann type, i.e.

$$q(\boldsymbol{x}) = \frac{\partial u(\boldsymbol{x})}{\partial n} = \bar{q}(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma_q, \tag{53}$$

where $u(\boldsymbol{x})$ is a unknown function, $b(\boldsymbol{x})$ is a known function, n is the unit outward normal to the boundary Γ, and $\bar{u}$ and $\bar{q}$ are the prescribed values of the function $u(\boldsymbol{x})$ and its normal derivative over the boundary Γ. Notice that $\Gamma = \Gamma_u \cup \Gamma_q$.

From the method of weighted residuals, we have the following integral equation:

$$\begin{aligned} &\int_\Omega (\nabla^2 u(\boldsymbol{x}) - b(\boldsymbol{x})) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Omega \\ &= \int_{\Gamma_q} (q(\boldsymbol{x}) - \bar{q}(\boldsymbol{x})) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma \\ &\quad - \int_{\Gamma_u} (u(\boldsymbol{x}) - \bar{u}(\boldsymbol{x})) q^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma, \end{aligned} \tag{54}$$

where $\boldsymbol{x}$ is the field point, ξ is the source point, u^* is chosen to be the fundamental solution to Poisson's equation (51), which means that

$$\nabla^2 u^* = -\delta(\xi, \boldsymbol{x}), \tag{55}$$

where $\delta(\xi, \boldsymbol{x})$ is Dirac δ function, and

$$q^*(\xi, \boldsymbol{x}) = \frac{\partial u^*(\xi, \boldsymbol{x})}{\partial n(\boldsymbol{x})}. \tag{56}$$

From Eq.(55), we have

$$u^* = \frac{1}{2\pi} \ln\left(\frac{1}{r}\right), \tag{57}$$

where

$$r(\xi, \boldsymbol{x}) = |\xi - \boldsymbol{x}|. \tag{58}$$

Integrating by parts for Eq.(54) and using the properties of the fundamental solution to Poisson's equation, we can obtain the following boundary integral equation

$$\begin{aligned} u(\xi) = &\int_\Gamma q(\boldsymbol{x}) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma - \int_\Gamma u(\boldsymbol{x}) q^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma \\ &- \int_\Omega b(\boldsymbol{x}) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Omega, \end{aligned} \tag{59}$$

when the source point ξ is located inside Ω, and

$$\begin{aligned} C(\xi) u(\xi) = &\int_\Gamma q(\boldsymbol{x}) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma \\ &- \int_\Gamma u(\boldsymbol{x}) q^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma \\ &- \int_\Omega b(\boldsymbol{x}) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Omega, \end{aligned} \tag{60}$$

when the source point ξ is located on boundary Γ. Here, $C(\xi)$ is the function of the internal angle that the boundary Γ makes at the given point ξ.

The boundary Γ can be separated into the sub-domains Γ_n, $n = 1, 2, \ldots, N$. Here, N is the total number of sub-domains. Then,

$$\Gamma = \bigcup_{n=1}^{N} \Gamma_n, \tag{61}$$

and Γ_n and Γ_{n-1} are connected with a point. Γ_n is not a boundary element, and thus no shape function is dependent upon it. Γ_n are only used so that the integrals in Eqs.(59) and (60) can be obtained numerically.

Equation (60) can then be written as

$$\begin{aligned} C(\xi) u(\xi) = &\sum_{n=1}^{N} \int_{\Gamma_n} q(\boldsymbol{x}) u^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma \\ &- \sum_{n=1}^{N} \int_{\Gamma_n} u(\boldsymbol{x}) q^*(\xi, \boldsymbol{x}) \mathrm{d}\Gamma \\ &+ \sum_{e=1}^{N_e} \left(\sum_{k=1}^{K} w_k (b u^*)_k \right) A_e. \end{aligned} \tag{62}$$

where w_k are the integration weight and the function (bu^*) needs to be evaluated at the k integration points,

K is the total number of integration points, N_e is the number of cells into which Ω has been divided and A_e is the area of each cell.

Some nodes are selected on each sub-domain Γ_n. The corresponding compact support domain for each node is built. The compact support domains must cover boundary Γ.

From the expression of the approximation function (45), we let

$$u(\boldsymbol{x}) = \sum_{I=1}^{n_I} \phi_I(\boldsymbol{x}) u_I, \tag{63}$$

$$q(\boldsymbol{x}) = \sum_{I=1}^{n_I} \phi_I(\boldsymbol{x}) q_I, \tag{64}$$

where $u_I = u(\boldsymbol{x}_I)$ and $q_I = q(\boldsymbol{x}_I)$, then equation (62) becomes

$$\begin{aligned} C(\xi^n)u(\xi^n) = &\sum_{n=1}^{N} \int_{\Gamma_n} u^*(\xi^n, \boldsymbol{x}) \sum_{I=1}^{n_I} \phi_I(\boldsymbol{x}) q(\xi^I) \mathrm{d}\Gamma \\ &- \sum_{n=1}^{N} \int_{\Gamma_n} q^*(\xi^n, \boldsymbol{x}) \sum_{I=1}^{n_I} \phi_I(\boldsymbol{x}) u(\xi^I) \mathrm{d}\Gamma + \sum_{e=1}^{N_e} \left(\sum_{k=1}^{K} w_k (bu^*)_k \right) A_e, \end{aligned} \tag{65}$$

where ξ^n are nodes, and n_I is the number of nodes in which the compact support domains cover the field point $\boldsymbol{x}$.

From Eq.(65) we then have

$$\begin{aligned} C(\xi^n)u(\xi^n) = &\sum_{n=1}^{N} \sum_{I=1}^{n_I} q(\xi^I) \int_{-1}^{1} u^*(\xi^n, \boldsymbol{x}) \phi_I(\boldsymbol{x}) J(\eta) \mathrm{d}\eta \\ &- \sum_{n=1}^{N} \sum_{I=1}^{n_I} u(\xi^I) \int_{-1}^{1} q^*(\xi^n, \boldsymbol{x}) \phi_I(\boldsymbol{x}) J(\eta) \mathrm{d}\eta + \sum_{e=1}^{N_e} \left(\sum_{k=1}^{K} w_k (bu^*)_k \right) A_e. \end{aligned} \tag{66}$$

where η is the local coordinate, and $J(\eta)$ is the Jacobian,

$$J(\eta) = \sqrt{\left(\frac{\mathrm{d}x_1}{\mathrm{d}\eta} \right)^2 + \left(\frac{\mathrm{d}x_2}{\mathrm{d}\eta} \right)^2}. \tag{67}$$

Using the numerical method for the integrals in Eq.(66), for a node ξ^n we have the following linear algebraic equations

$$\boldsymbol{B} + \boldsymbol{H}\boldsymbol{U} = \boldsymbol{G}\boldsymbol{Q}, \tag{68}$$

where

$$\boldsymbol{U} = (u_1, u_2, \cdots, u_N)^{\mathrm{T}}, \tag{69}$$

$$\boldsymbol{Q} = (q_1, q_2, \cdots, q_N)^{\mathrm{T}}, \tag{70}$$

and $\boldsymbol{H}$ and $\boldsymbol{G}$ are the corresponding coefficient matrices, and

$$\boldsymbol{B} = (B_1, B_2, \ldots, B_N)^{\mathrm{T}}, \tag{71}$$

$$B_i = \sum_{e=1}^{N_e} \left(\sum_{k=1}^{K} w_k (bu^*)_k \right) A_e. \tag{72}$$

Some singular integrals are included in Eq.(65), thus the logarithmic Gaussian quadrature formula and Cauchy principle value integral[29] are used when equation (68) is obtained.

Substituting the boundary conditions into Eq.(68) and solving the equations, we can obtain the unknowns at nodes on the boundary Γ.

When the source point ξ is located in domain Ω, from Eq.(59) we have

$$\begin{aligned} u(\xi^n) = &\sum_{n=1}^{N} \int_{\Gamma_n} u^*(\xi^n, \boldsymbol{x}) \sum_{I=1}^{n_I} \phi_I(\boldsymbol{x}) q(\xi^I) \mathrm{d}\Gamma \\ &- \sum_{n=1}^{N} \int_{\Gamma_n} q^*(\xi^n, \boldsymbol{x}) \sum_{I=1}^{n_I} \phi_I(\boldsymbol{x}) u(\xi^I) \mathrm{d}\Gamma \\ &+ \sum_{e=1}^{N_e} \left(\sum_{k=1}^{K} w_k (bu^*)_k \right) A_e, \end{aligned} \tag{73}$$

and we can then obtain the unknowns at point ξ.

This is the improved boundary element-free method (IBEFM) for two-dimensional potential problems.

5. Example problems

Three examples are selected to demonstrate the applicability of the BEFM to two-dimensional potential problems. The results that are obtained from these examples are compared with the existing analytical solutions that have been published in the literature.

The first example that is considered is a temperature field of a rectangular plate governed by Laplace's equation

$$\nabla^2 T = \frac{\partial^2 T}{\partial x_1^2} + \frac{\partial^2 T}{\partial x_2^2} = 0,$$
$$x_1 \in [0,5], \quad x_2 \in [0,10].$$

And the boundary conditions are

$$T(x_1,0) = 0, \quad T(0,x_2) = 0,$$
$$T(x_1,10) = 100\sin(\pi x_1/10),$$

and

$$\frac{\partial T(5,x_2)}{\partial x_1} = 0.$$

The analytical solution of this problem is

$$T(x_1,x_2) = \frac{100\sin(\pi x_1/10)\mathrm{sh}(\pi x_2/10)}{\mathrm{sh}(\pi)}.$$

The node distribution on the rectangular plate is shown in Fig.1. In this analysis, we employed 46 nodes on the boundary. The numerical results are shown in Fig.2. It is evident that the results accord well with the analytical solution.

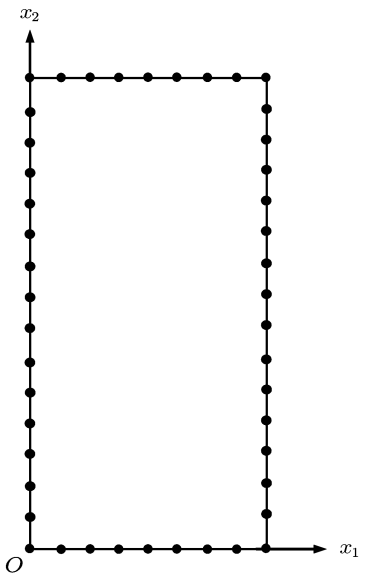


Fig.1. Node distribution.

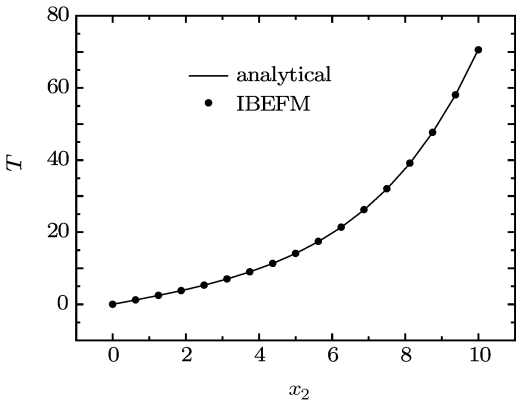


Fig.2. Analytical and numerical solutions for the temperature at $x_1 = 2.5$.

The second example we considered is a temperature field of a rectangular plate governed by Poisson's equation

$$\nabla^2 T(x_1,x_2) = -2(x_1 + x_2 - x_1^2 - x_2^2),$$
$$0 \le x_1 \le 1, \quad 0 \le x_2 \le 1.$$

And the boundary condition is

$$T(x_1,x_2) = 0.$$

The analytical solution of this problem is

$$T(x_1,x_2) = (x_1 - x_1^2)(x_2 - x_2^2).$$

The node distribution on the rectangular plate is shown in Fig.3. In this analysis, we employed 32 nodes on the boundary. The numerical results are shown in Fig.4. It is again evident that the present results are in excellent agreement with the analytical solution.

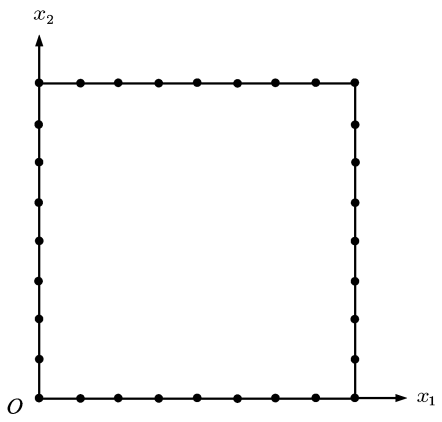


Fig.3. Node distribution.

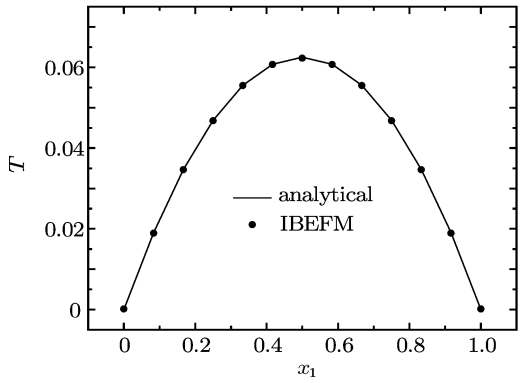


Fig.4. Analytical and numerical solutions of the temperature at $x_2 = 0.5$.

The third example we considered is a temperature field of a circular plate governed by Laplace's equation

$$\nabla T^2 = 0.$$

And the boundary condition is

$$T\,|_{r=r_0} = r_0^2 \cos\theta \sin\theta.$$

where r_0 is the radius of the circular plate, and $r_0 = 1$.

The analytical solution of this problem is

$$u(r,\theta) = \frac{1}{2} r^2 \sin(2\theta).$$

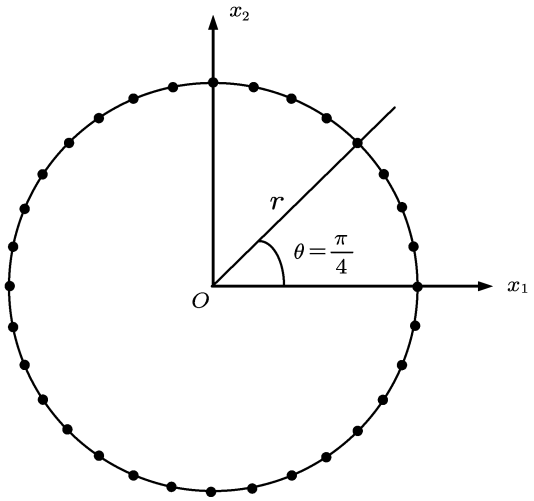


Fig.5. Node distribution.

The node distribution on the rectangular plate is shown in Fig.5. In this analysis, we employed 32 nodes on the boundary. The numerical results are shown in Fig.6. It can be seen again that the present results are in excellent agreement with the analytical solution.

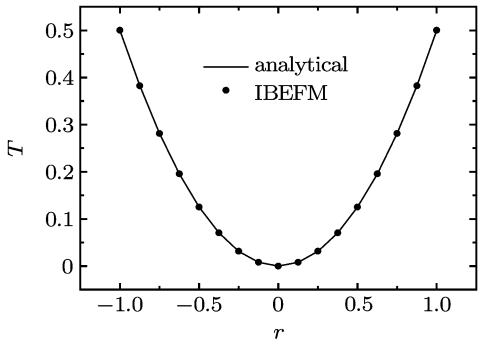


Fig.6. Analytical and numerical solutions of the temperature at $\theta = \pi/4$.

6. Conclusions

The interpolating moving least-squares (IMLS) method, in which the shape function satisfies the property of the Kronecker δ function, is discussed, and the corresponding formulae of the IMLS method are revised in this paper. Then, on the basis of the boundary element-free method (BEFM), combining the boundary integral equation method with the IMLS method, the improved boundary element-free method (IBEFM) of solving two-dimensional potential problems is presented. The IBEFM is a direct meshless boundary integral equation method in which the basic unknown quantity is the real solution of the nodal variables. In the BEFM and the IBEFM, the essential boundary conditions are applied directly and easily. But in the IBEFM, because the IMLS method is used, the essential boundary conditions are applied directly without doubt. Thus the IBEFM gives a greater computational precision. The IBEFM is a perfect direct meshless boundary integral equation method.

References

[1] Belytschko T, Krongauz Y, Organ D, Fleming M and Krysl P 1996 *Comp. Meth. Appl. Mech. Eng.* **139** 3

[2] Li S F and Liu W K 2002 *Appl. Mech. Rev.* **55** 1

[3] Li S C, Cheng Y M and Li S C 2006 *Acta Phys. Sin.* **55** 4760 (in Chinese)

[4] Cheng Y M and Li J H 2005 *Acta Phys. Sin.* **54** 4463 (in Chinese)

[5] Cheng R J and Cheng Y M 2007 *Acta Phys. Sin.* **56** 5569 (in Chinese)

[6] Chen L and Cheng Y M 2008 *Acta Phys. Sin.* **57** 1 (in Chinese)

[7] Cheng R J and Cheng Y M 2008 *Acta Phys. Sin.* **57** 6037 (in Chinese)

[8] Chen L and Cheng Y M 2008 *Acta Phys. Sin.* **57** 6047 (in Chinese)

[9] Mukherjee Y X and Mukherjee S 1997 *Int. J. Numer. Meth. Eng.* **40** 797

[10] Kothnur V S, Mukherjee S and Mukherjee Y X 1999 *Int. J. Numer. Meth. Eng.* **46** 1129

[11] Chati M K, Mukherjee S and Mukherjee Y X 1999 *Int. J. Numer. Meth. Eng.* **46** 1163

[12] Chati M K and Mukherjee S 2000 *Int. J. Numer. Meth. Eng.* **47** 1523

[13] Zhu T, Zhang J and Atluri S N 1999 *Engineering Analysis with Boundary Elements* **23** 375

[14] Atluri S N, Sladek J, Sladek V and Zhu T 2000 *Comput. Mech.* 25 180

[15] Sladek V, Sladek J, Atluri S N and Keer R V 2000 *Comput. Mech.* **25** 394

[16] Dai B D and Cheng Y M 2007 *Acta Phys. Sin.* **56** 597 (in Chinese)

[17] Liew K M, Cheng Y and Kitipornchai S 2006 *Int. J. Numer. Meth. Eng.* **65** 1310

[18] Kitipornchai S, Liew K M and Cheng Y M 2005 *Comput. Mech.* **36** 13

[19] Cheng Y and Peng M 2005 *Sci. Chin.* Ser. G *Phys., Mech. & Astro.* **48** 641

[20] Liew K M, Cheng Y M and Kitipornchai S 2005 *Int. J. Numer. Meth. Eng.* **64** 1610

[21] Sun Y, Zhang Z, Kitipornchai S and Liew K M 2006 *Int. J. Eng. Sci.* **44** 37

[22] Liew K M, Cheng Y M and Kitipornchai S 2007 *Int. J. Solids Struc.* **44** 4220

[23] Liew K M, Sun Y and Kitipornchai S 2007 *Int. J. Numer. Meth. Eng.* **69** 729

[24] Qin Y X and Cheng Y M 2006 *Acta Phys. Sin.* **55** 3215 (in Chinese)

[25] Peng M and Cheng Y M 2009 *Engineering Analysis with Boundary Elements* **33** 77

[26] Lancaster P and Salkauskas K 1981 *Math. Comput.* **37** 141

[27] Zhang Z, Liew K M and Cheng Y M 2008 *Engineering Analysis with Boundary Elements* **32** 100

[28] Zhang Z, Liew K M, Cheng Y M and Lee Y Y 2008 *Engineering Analysis with Boundary Elements* **32** 241

[29] Brebbia C A, Telles J C F and Wrobel L C 1984 *Boundary Element Techniques Theory and Applications in Engineering* (Berlin: Springer-Verlag)