A double-layer interpolation method for implementation of BEM analysis of problems in potential theory

Jianming Zhang*, Weicheng Lin, Yunqiao Dong, Chuanming Ju

State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

Correspondence to: Jianming Zhang

College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China

Telephone: +86-731-88823061

E-mail: zhangjm@hnu.edu.cn

Abstract

A double-layer interpolation method (DLIM) is proposed to improve the performance of the boundary element method (BEM). In the DLIM, the nodes of an element are sorted into two groups: (i) nodes inside the element, called source nodes, and (ii) nodes on the vertices and edges of the element, called virtual nodes. With only source nodes, the element becomes a conventional discontinuous element. Taking into account both source and virtual nodes, the element becomes a standard continuous element. The physical variables are interpolated by continuous elements (first-layer interpolation), while the boundary integral equations are collocated at the source nodes only. We further established additional constraint equations between source and virtual nodes using a moving least-squares (MLS) approximation (second-layer interpolation). Using these constraints, a square coefficient matrix of the overall system of linear equations was finally achieved. The DLIM keeps the main advantages of MLS, such as significantly alleviating the meshing task, while providing much better accuracy than the traditional BEM. The method has been used successfully for solving potential problems in two dimensions. Several numerical examples in comparison with other methods have demonstrated the accuracy and efficiency of our method.

Key words: Boundary element method, double layer interpolation method, moving least-squares approximation

1. Introduction

The boundary integral equation method (BIEM) has been studied widely with regard to problems in potential theory [1-6]. The method is classified as a boundary-type method, which can reduce the dimensions of the original problem by one. With advances in computer technology, the boundary element method (BEM) has become a well-established method to deal with practical engineering problems [7-9] involving, for example, heat conduction [10, 11], acoustics [12, 13], linear elasticity [14], and crack formation [15]. Compared with the finite element method (FEM), the BEM offers several advantages, including higher accuracy and only discretizing the boundary rather than the entire domain. In contrast with FEM, in which the trial functions must have at least C^0 continuity, the BEM has no continuity requirement for the field variables at the interfaces between elements. Thus, both continuous and discontinuous elements can be used with BEM [16, 17].

A discontinuous element, with all source nodes inside the element, provides many advantages, with regard to simplifying mesh generation, the assembly and solution of system equations, and the evaluation of hypersingular integrals [18]. However, using discontinuous elements, the C^0 continuity for the potential on the whole boundary and for the boundary normal flux on the smooth parts of the boundary is not guaranteed. Moreover, for the same level of accuracy, many more nodes are required. This means more computer resources and CPU time, because the size of the final system of the coefficient matrix increases significantly. Continuous elements can guarantee C^0 continuity, but not $C^{1,\alpha}$ continuity, which is necessary for the hypersingular integral equation [19]. In particular, continuous elements are inconvenient and inaccurate when used to interpolate discontinuous fields, such as normal fluxes at corners [20] and edges of a body.

To escape this dilemma and to unify the approximation of both continuous and discontinuous fields, a double-layer interpolation method (DLIM) is described here. The method is achieved by combining conventional polynomial element interpolation and moving least-squares (MLS) approximation.

In the DLIM, an element is constructed by adding virtual nodes to a conventional discontinuous element at the vertices and edges of a geometric element. This element is equivalent to a standard continuous element. For example, two virtual nodes together with one source node in the conventional constant element comprise an entire continuous quadratic element for two-dimensional (2D) problems. When using these continuous elements to interpolate the boundary field variables (first-layer interpolation), the interpolation accuracy will increase by two orders compared with the corresponding discontinuous element. Furthermore, by manipulating the influence domains of some virtual nodes in the MLS (second-layer interpolation), both continuous and discontinuous fields can be readily and accurately approximated. Because the BIEs are not collocated at the virtual nodes, the evaluation of hypersingular integrals and the treatment of geometrical and physical corners can be simplified. Thus, the DLIM is able to unify the continuous and discontinuous element interpolation methods. Based on the features, it can be readily extended to solve problems involving discontinuity, such as crack propagation and contact problems [21, 22].

For a three-dimensional (3D) problem, even if we use the discontinuous grids, both continuous and discontinuous fields can still be accurately approximated by using the DLIM. Due to the fact that it is easier to obtain a discontinuous grid than a continuous grid (see Fig. 1), we can expect that our method will substantially simplify the mesh generation. This is a major advantage of the DLIM for solving problems in three-dimensions.

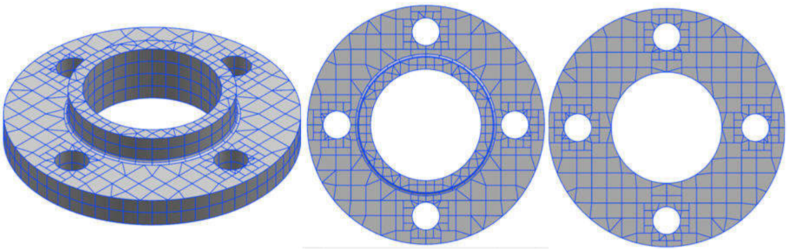


Fig.1. The discontinuous surface girds of the flange plate.

In this paper, we present a general formulation of the double-layer interpolation boundary element method (DLI-BEM). The boundary integral equations (BIEs) in this method are collocated at the source nodes only. This scheme leads to the number of interpolation nodes being more than the number of linear algebraic equations obtained by the discretized BIE. Thus, we must provide additional constraint equations to make the final system of equations solvable. These equations are obtained using the MLS approximation to construct relationships between source and virtual nodes. With these relationships, a square matrix for the final overall system of linear algebraic equations can be achieved. Obviously, the DLI-BEM requires much more CPU seconds for computing and assembling these coefficient matrices than the traditional BEM. For the same level of accuracy, however, the DLI-BEM still gains efficiency.

In the DLI-BEM, the MLS is used only for assembling the coefficient matrices and there is no need to calculate the derivatives of the MLS shape functions. In the boundary integration process, variables at Gaussian points are evaluated by the polynomial shape functions of the DLIM elements, rather than the MLS approximation. Moreover, in the MLS computation, the source nodes covered in each influence domain of virtual nodes can be obtained directly by searching the neighboring cells instead of searching all source nodes. Thus, the computational efficiency of the MLS in the DLI-BEM is much higher than that of MLS constructed by scattered nodes in a pure mesh-free method.

The MLS, proposed by Lancaster and Salkauskas [23], was first introduced into computational mechanics by Belytschko [24] for a mesh-less method. However, the original MLS did not have a delta function property, which leads to difficulties in imposing boundary conditions. To restore the delta function property, Lancaster and Salkauskas [23] developed an interpolating moving least-square (IMLS) method, in which only some specific singular weight functions are used. Later, Wang et al. proposed an improved interpolating moving least-square (IIMLS) method [25], which not only satisfies the property of a delta function but can be used with any kind of weight function. Li [26] theoretically proved the delta function property of the IIMLS shape function. In this paper, we will adopt the IIMLS for the second-layer interpolation.

This paper is organized as follows. In Section 2, the double-layer interpolation method is described in detail. Section 3 describes the formulation of the double-layer interpolation boundary element method (DLI-BEM) for potential problems. Numerical examples for an interpolation test and several 2-D potential problems are presented in Section 4. The paper ends with our conclusions in Section 5.

2. The double-layer interpolation method

In this paper, we begin by considering problems in two dimensions. For a 2D domain, the boundary elements are in one dimension. In this section, we describe the double-layer interpolation method (DLIM) for problems in two dimensions.

2.1 The elements of DLIM

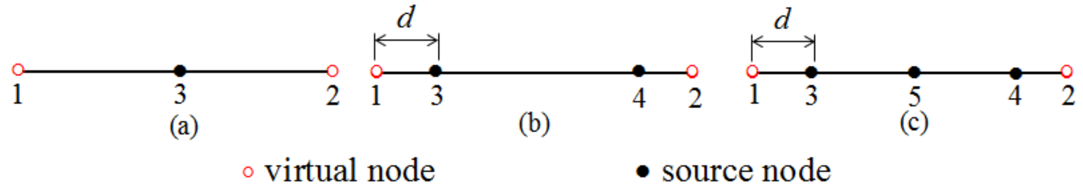


Fig. 2. Elements of the double-layer interpolation method (DLIM) for two-dimensional (2D) problems: (a) constant, (b) linear, and (c) quadratic.

The elements of DLIM include continuous and discontinuous elements. Figure 2 shows three kinds of DLIM elements with different orders. The two end nodes shown by void circles are referred to as virtual nodes, and the internal nodes shown by solid discs are source nodes. Ignoring the virtual nodes, the elements are conventional discontinuous elements. The virtual nodes and the source nodes together comprise an entire element; this element is equivalent to a conventional continuous element. For example, the DLIM element shown in Fig. 1(a) is constructed by one source node and two virtual nodes. The shape functions of the new elements (a), (b), and (c) are given by Eqs. (1)–(3), respectively.

$$\begin{cases} N_1 = 0.5\xi(\xi - 1) \\ N_2 = 0.5\xi(\xi + 1) \\ N_3 = (1 - \xi)(1 - \xi) \end{cases} , \tag{1}$$

$$\begin{cases} N_1 = -\dfrac{[\xi + (1-d)][\xi - (1-d)](\xi - 1)}{2d(2-d)} \\ N_2 = \dfrac{[\xi + (1-d)][\xi - (1-d)](\xi + 1)}{2d(2-d)} \\ N_3 = \dfrac{[\xi - (1-d)](\xi + 1)(\xi - 1)}{2d(1-d)(2-d)} \\ N_4 = -\dfrac{[\xi + (1-d)](\xi + 1)(\xi - 1)}{2d(1-d)(2-d)} \end{cases}, \tag{2}$$

$$\begin{cases} N_1 = \dfrac{[\xi + (1-d)][\xi - (1-d)](\xi - 1)\xi}{2d(2-d)} \\ N_2 = \dfrac{[\xi + (1-d)][\xi - (1-d)](\xi + 1)\xi}{2d(2-d)} \\ N_3 = -\dfrac{[\xi - (1-d)](\xi + 1)(\xi - 1)\xi}{2d(2-d)(1-d)^2} \\ N_4 = -\dfrac{[\xi + (1-d)](\xi + 1)(\xi - 1)\xi}{2d(2-d)(1-d)^2} \\ N_5 = \dfrac{[\xi + (1-d)][\xi - (1-d)](\xi + 1)(\xi - 1)}{(1-d)^2} \end{cases}, \tag{3}$$

where $\xi \in [-1,1]$ is the natural coordinate defined on the element, and $d \in (0,1)$ representing the offset of source nodes. In this paper, the value of d is taken to be 0.25. From Eqs. (1)–(3), the interpolation accuracy of DLIM elements increases by two orders compared with the corresponding discontinuous element.

We will use the DLIM element to interpolate boundary variables (this is referred as first-layer interpolation), but we will collocate the BIE only at the source nodes (which is why we call them source nodes). In this way, the number of linear algebraic equations obtained by discretizing the BIE will be less than the number of interpolation nodes. Thus, we must provide additional constraint equations to make the final system of equations solvable. These equations will be obtained by the second-layer interpolation using the moving least-squares (MLS) approximation to construct relationships between variables on source nodes and virtual nodes. We will use these constraints to condense the degrees for all virtual nodes. The second-layer interpolation will be described in the next subsection.

2.2 The moving least-squares method

The MLS approximation is considered the second-layer interpolation here. In the proposed method, the MLS approximation is used just for constructing the relationship between the source nodes and the virtual nodes, rather than evaluating the shape functions at each Gaussian point in the numerical integration. There is no need to evaluate the derivatives of the MLS shape functions. The MLS approximation is performed independently on piecewise smooth segments Γ_i (*i* = 1, 2, …, *n*), which naturally consist of the boundary.

The discussion below addresses a field function *f*(*t*) on a one-dimensional (1-D) boundary of a 2-D body. The function values at virtual nodes approximated by the MLS [23, 24] are defined as

$$f(t^v) = \sum_{j=1}^{M} \phi_j(t^v) \hat{f}_j^s , \tag{4}$$

where t^v is a curvilinear co-ordinates of a virtual node on Γ_i, respectively. *M* denotes the total number of source nodes of which the influence domain covers the virtual node t^v and $\hat{f}_j^s$ are the fictitious source nodal values, and $\phi_j(t^v)$ represents the shape function of the MLS corresponding to source node t_j^s, which is given by

$$\phi_j(t^v) = \sum_{k=1}^{m} p_k(t^v)[\mathbf{A}^{-1}(t^v)\mathbf{B}(t^v)]_{kj}, \qquad m = 4 , \tag{5}$$

with matrices **A**(t_I^v) and **B**(t_I^v), defined by

$$\mathbf{A}(t^v) = \sum_{j=1}^{M} w_j(t^v, t_j^s)\mathbf{p}(t_j^s)\mathbf{p}^{\mathrm{T}}(t_j^s) ,$$

$$\mathbf{B}(t^v) = [w_1(t^v, t_1^s)\mathbf{p}(t_1^s), w_2(t^v, t_2^s)\mathbf{p}(t_2^s), \ldots, w_M(t^v, t_M^s)\mathbf{p}(t_M^s)] ,$$

where $w_j(t^v, t_j^s)$ (*j* = 1, 2, …, *M*) is the weight function at the source node t_j^s, and $p_k(t^v)$ are monomial basis functions of virtual node t^v, and $\mathbf{p}^{\mathrm{T}}(t_j^s)$ (*j* = 1, 2, …, *M*) is a basis function vector of source node t_j^s. In the numerical implementation presented later in this paper, a cubic basis is used:

$$\mathbf{p}^{\mathrm{T}}(\bar{t}) = [1, \bar{t}, \bar{t}^2, \bar{t}^3], \qquad m = 4. \tag{6}$$

where $\bar{t}$ can either be the virtual node t^v or the source node t_j^s.

Because the shape functions in Eq. (5) lack of the property of a delta function, an IIMLS [25] was proposed. With a set of new basis functions $\tilde{p}_k(t^v, \bar{t})$, the IIMLS interpolants for $f(t^v)$ are defined as

$$f(t^v) = \sum_{j=1}^{M} \Phi_j(t^v) f_j^s, \tag{7}$$

where f_j^s are the function values of source nodes. The shape function $\Phi_j(t^v)$ of the IIMLS is given by

$$\Phi_j(t^v) = \sum_{k=2}^{m} \tilde{p}_k(t^v)[\mathbf{C}^{-1}(t^v)\mathbf{D}(t^v)]_{kj} + g(t^v, t_j^s), \qquad m = 4, \tag{8}$$

with matrices $\mathbf{C}(t^v)$ and $\mathbf{D}(t^v)$ defined by

$$\mathbf{C}(t^v) = \sum_{j=1}^{M} w_j(t^v, t_j^s)\tilde{\mathbf{p}}(t^v, t_j^s)\tilde{\mathbf{p}}^{\mathrm{T}}(t^v, t_j^s),$$

$$\mathbf{D}(t^v) = \sum_{j=1}^{M} w_j(t^v, t_j^s)\tilde{\mathbf{p}}(t^v, t_j^s)[\mathbf{e}_j - \mathbf{g}(t^v)],$$

$\mathbf{e}_j$ is a M-unit row vector with the j^{th} component being 1, and

$$\tilde{\mathbf{p}}^{\mathrm{T}}(t^v, t_j^s) = [\tilde{p}_2(t^v, t_j^s), \tilde{p}_3(t^v, t_j^s), \tilde{p}_4(t^v, t_j^s)],$$

$$\mathbf{g}(t^v) = [g(t^v, t_1^s), g(t^v, t_2^s), ..., g(t^v, t_M^s)].$$

The new basis functions $\tilde{p}_k(t^v, \bar{t})$ are defined as

$$\tilde{p}_k(t^v, \bar{t}) = p_k(\bar{t}) - \sum_{j=1}^{M} g(t^v, t_j^s) p_k(t_j^s), \qquad k = 1, 2, \ldots, m, \tag{9}$$

with $g(t^v, t_j^s)$ is given by

$$g(t^v, t_j^s) = \frac{\eta(t^v, t_j^s)}{\sum_{i=1}^{M} \eta(t^v, t_j^s)}, \qquad j = 1, 2, \ldots, M,$$

$$\eta(t^v, t_j^s) = \prod_{i=1, i\neq j}^{M} \frac{\left\| t^v - t_j^s \right\|^2}{\left\| t_i^s - t_j^s \right\|^2}, \qquad j = 1, 2, \ldots, M \,.$$

Equations (5) and (8) are shape functions of the MLS and IIMLS approximations, respectively, corresponding to source node t_j^s. The MLS and IIMLS approximations are well defined when matrices $\mathbf{A}(t^v)$ in Eq. (5) and $\mathbf{C}(t^v)$ in Eq. (8) are invertible.

For implementing the MLS and IIMLS, the weight functions should be chosen first. As pointed out in Ref. [24], a Gaussian-type weight function yields excellent results. Thus, in our work, we used the following weight functions:

$$w_j(t^v, t_j^s) = \begin{cases} \dfrac{e^{-(d_j/c_j)^2} - e^{-(\hat{d}_j/c_j)^2}}{1 - e^{-(\hat{d}_j/c_j)^2}}, & 0 \le d_j \le \hat{d}_j \\ 0, & d_j \ge \hat{d}_j \end{cases}, \tag{10}$$

where $d_j = \left| t^v - t_j^s \right|$ is the distance between a virtual node t^v and a source node t_j^s measured along Γ_i, c_j is a constant controlling the shape of the weight function, and $\hat{d}_j$ is the size of the support for the weight function w_j and determines the support of the source node t_j^s. To guarantee the regularity of matrices $\mathbf{A}(t^v)$ and $\mathbf{C}(t^v)$, the choice of $\hat{d}_j$ should be sufficiently large to have at least m and $m-1$ source nodes, respectively, of which the influence domain covers the virtual node t^v.

2.3 Approximation of continuous and discontinuous fields

Figure 3 shows a rectangular domain discretized by 9 DLIM elements with 21 source nodes and 14 virtual nodes. These elements are used to interpolate potentials and normal fluxes on the boundary. At sharp corners and points at which the boundary condition is discontinuous, we put two virtual nodes (nodes 23 and 24, for example), one for each element. At a point where two adjacent elements are smoothly connected, we put only one virtual node (node 27, for example), letting the two elements share a same virtual node, thus naturally leading to continuity at the point. Alternatively, we can also put two virtual nodes at the point, making all elements independent of each other. In this case, for continuous fields, a continuity constraint is added.

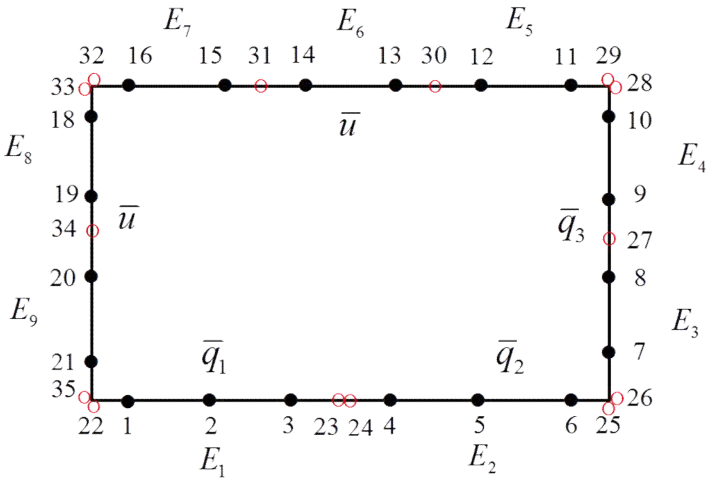


Fig. 3. Boundary discretized by the DLIM elements.

The visibility criterion [24] will be adopted here to approximate the continuous and discontinuous fields. The difference between the proposed method and the original MLS is that in the proposed method, the source nodes covered in the influence domain of a virtual node are obtained by searching the neighboring cells of the virtual node instead of searching all source nodes. This feature makes our method much more efficient than the original MLS.

With the visibility criterion, the influence domains of some specific nodes may be different for approximating the continuous and discontinuous fields. When approximating the discontinuous fields, the virtual nodes and the source nodes beyond a sharp corner or a discontinuous point are considered as opaque in the weight functions of MLS. For approximating the continuous fields, on the other hand, all nodes in the neighborhood of an evaluating point are taken as transparent. In the construction of an influence domain of a virtual node, a projecting line (in curve parametric space) from a virtual node to a cell is imagined to be a path of light. If the path encounters an opaque node, it is terminated and the source nodes of the cell are excluded in the influence domain of this virtual node. An example is given in Fig. 4.

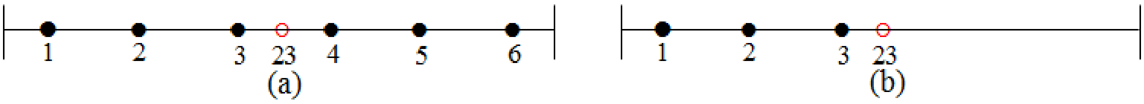


Fig. 4. Influence domains of virtual node 23: (a) for interpolating the potential and (b) for interpolating the normal flux.

In Fig. 4(a), for interpolating the potential (a continuous field), the source nodes in the influence domain of virtual node 23 are obtained by searching the left and right connecting cells. In case the normal flux is not continuous at node 23, another node 24 is added at the same location. Then, node 24 is an opaque node, and the source nodes covered in the influence domain of node 23 are nodes 1, 2, and 3 (see Fig. 4(b)).

For virtual nodes on smooth parts of the boundary (e.g. node 30 in Fig. 3), the potential u_i^v and the normal flux q_i^v are approximated with Eqs. (11) and (12):

$$u_i^v = \sum_{j=1}^{M^i} \Phi_j(t_i^v) u_j^s \tag{11}$$

$$q_i^v = \sum_{j=1}^{M^i} \Phi_j(t_i^v) q_j^s \tag{12}$$

where t_i^v is a curvilinear co-ordinates of i^{th} virtual node, and $\Phi_j(t^v)$ is the shape function at source node t_j^s (e.g., Eq. (10)), M^i denotes the total number of source nodes, u_j^s and q_j^s are the nodal values for the potential and normal flux at the source node t_j^s, which is covered in the influence domain of virtual node t_i^v.

For virtual nodes at sharp corners (e.g., nodes 24 and 25 in Fig. 3), q_i^v is still approximated by Eq. (15). u_i^v is approximated by

$$u_i^v = \sum_{l=1}^{L} \sum_{j=1}^{M^l} \omega_l(t_{i(l)}^v) \Phi_j(t_{i(l)}^v) u_{j(l)}^s, \qquad (L = 2 \text{ in 2D problem}) \tag{13}$$

with

$$\begin{cases} \omega_1(t^v_{i(2)}) = \dfrac{\min\limits_{1\le j\le M^2}\left\{\left|t^v_{i(2)} - t^s_{j(2)}\right|\right\}}{\sum\limits_{k=1}^{L}\min\limits_{1\le j\le M_l}\left\{\left|t^v_{i(k)} - t^s_{j(k)}\right|\right\}} \\ \omega_2(t^v_{i(1)}) = \dfrac{\min\limits_{1\le j\le M^1}\left\{\left|t^v_{i(1)} - t^s_{j(1)}\right|\right\}}{\sum\limits_{k=1}^{L}\min\limits_{1\le j\le M_l}\left\{\left|t^v_{i(k)} - t^s_{j(k)}\right|\right\}} \end{cases}$$

where L is a number of adjacent edge of i^{th} virtual node t^v_i on element, $t^v_{i(l)}$ is the curvilinear coordinate of virtual node t^v_i on the l^{th} adjacent edge, M^l is the total number of source nodes covered in the influence domain of $t^v_{i(l)}$, $t^s_{j(k)}$ is the curvilinear coordinate at the j^{th} source node covered in the influence domain of $t^v_{i(k)}$, and $\left|t^v_{i(k)} - t^s_{j(k)}\right|$ denotes the arc length between $t^v_{i(k)}$ and $t^s_{i(k)}$ along each edge Γ_k.

In the DLIM, the potential u and the normal flux q at any boundary point P are approximated by the first-layer interpolation:

$$u(P) = u(\xi) = \sum_{i=1}^{2} N_i(\xi)u^v_i + \sum_{i=3}^{K} N_i(\xi)u^s_i\ , \tag{14}$$

$$q(P) = q(\xi) = \sum_{i=1}^{2} N_i(\xi)q^v_i + \sum_{i=3}^{K} N_i(\xi)q^s_i\ , \tag{15}$$

where $\xi \in [-1,1]$ is the natural coordinate of the point P defined on the element, and $N_i(\xi)$ represents the shape function of the i^{th} interpolation node in each element (e.g., Eqs. (1)–(3)), K is equal to 3 (4 or 5) according to the order of the DLIM element (see Fig. 1), and u^s_i and q^s_i represent nodal values of potential and normal flux at i^{th} source nodes in element, respectively. u^v_i and q^v_i are nodal values of potential and normal flux at i^{th} virtual node on element, which are determined by the second-layer interpolation (e.g., Eqs. (11)-(13)).

Substituting Eq. (12) into Eq. (15) yields the DLIM formulation for the normal flux,

$$q(\xi) = \sum_{i=1}^{2}\sum_{j=1}^{M^i} N_i(\xi)\Phi_j(t^v_i)q^s_j + \sum_{i=3}^{K} N_i(\xi)q^s_i\ , \tag{16}$$

For DLIM elements without sharp corners and discontinuous boundary conditions (e.g., element 6 in Fig. 3), substituting Eq. (11) into Eq. (14) yields the DLIM formulation for the potential

$$u(\xi)=\sum_{i=1}^{2}\sum_{j=1}^{M^i}N_i(\xi)\Phi_j(t_i^v)u_j^s+\sum_{i=3}^{K}N_i(\xi)u_i^s\ , \tag{17}$$

For DLIM elements with a sharp corners or discontinuous boundary condition (e.g., elements 1 and 2 in Fig. 3), substituting Eq. (13) into Eq. (14) yields the DLIM formulation for the potential

$$u(\xi)=\sum_{i=1}^{2}\sum_{l=1}^{L^i}\sum_{j=1}^{M^l}N_i(\xi)\omega_l(t_{i(l)}^v)\Phi_j(t_{i(l)}^v)u_{j(l)}^s+\sum_{i=3}^{K}N_i(\xi)u_i^s\ . \tag{18}$$

where L^i is a number of adjacent edge of i^{th} virtual node t_i^v on element.

From the Eqs. (11)-(18), It is seen that, by manipulating the influence domains of some virtual nodes in the second-layer interpolation (see Fig. 4), the DLIM is able to accurately approximate both continuous and discontinuous fields, as well as readily guarantee the C^0 continuity for continuous fields. This is done by putting two virtual nodes at a sharp corner or a point where the boundary condition is discontinuous (see Fig. 3). As the BIEs are not collocated at virtual nodes, the proposed method retains the advantages of discontinuous element, such as simplifying the evaluation of hypersingular integrals and the treatment of geometric corners and discontinuous boundary conditions. Therefore, the DLIM unifies the continuous and discontinuous element interpolation methods, and is a general method for solving the problems involving discontinuity.

2.4 Algorithm of DLIM

The procedure of the DLIM is as follows:

1. Find an element *E* which includes a given boundary point *P*.
2. Calculate physical variables at virtual nodes of the element *E* by the second-layer interpolation (e.g., Eqs. (11)-(13)).
3. Calculate physical variables at point *P* by the first-layer interpolation (e.g., Eqs. (14) and (15)).

From the procedure, the accuracy of the DLIM is affected by both the first-layer and the second-layer interpolations. By adding two virtual nodes to a conventional discontinuous element at the vertices, the original discontinuous element becomes a high-order continuous element (see Fig. 2) so that the interpolation accuracy of the DLIM elements is improved (first -layer interpolation). Furthermore, the physical variables at all virtual nodes can be accurately approximated by the MLS or IIMLS approximations (second-layer interpolation). Thus, the DLIM combines the advantages of the stability of element interpolation and the high accuracy of the MLS or IIMLS, while overcoming their disadvantages. This will be verified by numerical examples in Section 4.

3. The double-layer interpolation boundary element method

3.1 Boundary integral equation

The well-known boundary integral equation (BIE) for the potential problem with a 2D finite region Ω bounded by a boundary Γ is

$$c(P)u(P)=\int_{\Gamma}G(P,Q)q(Q)d\Gamma(Q)-\int_{\Gamma}\frac{\partial G(P,Q)}{\partial n(Q)}u(Q)d\Gamma(Q),\qquad P,Q\in\Gamma \tag{19}$$

where *u* and *q* are the potential and normal flux of the field function on the boundary Γ, respectively, *n* is a unit-normal pointing outward of Ω at the field point *Q*, the coefficient *c*(*P*) =1, 1/2 or 0 when the source point *P* is in the interior region Ω, on smooth boundary Γ or in the exterior region, and *G*(*P*,*Q*) is Green's function in 2D space given by

$$G(P,Q)=-\frac{1}{2\pi}\ln r, \tag{20}$$

where $r=\left|P-Q\right|$.

3.2 Discretization of the BIE using the DLIM elements

The boundary Γ of the 2D domain Ω will be discretized by the DLIM elements (Fig. 2). The total numbers of elements, source nodes, and field nodes are *E*, *N*, and *M*, respectively. Thus, the discretization form of Eq. (20) can be written as

$$\sum_{j=1}^{E}\sum_{\alpha=1}^{M^j}h_{ij}^{\alpha}u_j^{\alpha}=\sum_{j=1}^{E}\sum_{\alpha=1}^{M^j}g_{ij}^{\alpha}q_j^{\alpha},\qquad i=1,2,\ ...\ ,N. \tag{21}$$

with

$$h_{ij}^{\alpha}=\int_{\Gamma_j}\frac{\partial G(P_i,Q)}{\partial n(Q)}N_j^{\alpha}(Q)d\Gamma(Q)+\frac{1}{2}\delta_{ij}^{\alpha},$$

$$g_{ij}^{\alpha}=\int_{\Gamma_j}G(P_i,Q)N_j^{\alpha}(Q)d\Gamma(Q),$$

where u_j^{α} and q_j^{α} are the potential and the normal flux of the α th interpolation node of the jth element, respectively, M^j denotes the total number of interpolation nodes of the jth element, and $N_j^{\alpha}(Q)$ represents the shape function of the first-layer interpolation (e.g., Eqs. (1)–(3)) for the α th interpolation node of the jth element.

The matrix form of Equation (21) is

$$\mathbf{Hu}=\mathbf{Gq}, \tag{22}$$

where **H** and **G** are $N\times M$ coefficient matrices, and **u** and **q** are vectors of M components containing potentials and normal fluxes of all interpolation nodes, respectively.

In the DLI-BEM, the BIEs are collocated at the source nodes only. This will lead to the number of linear algebraic equations being less than the number of interpolation nodes. To make the final system of equations solvable, additional constraint equations should be applied. These constraints will be used to condense the degrees for all virtual nodes, which are explained in the next subsection.

3.3 Matrix assembly and solution

Distinguishing the source and virtual nodes, Eq. (22) can be rewritten as

$$\begin{bmatrix}\bar{\mathbf{H}}^s & \mathbf{H}^s & \bar{\mathbf{H}}^v & \mathbf{H}_1^v & \mathbf{H}_2^v\end{bmatrix}\begin{bmatrix}\bar{\mathbf{u}}^s\\ \mathbf{u}^s\\ \bar{\mathbf{u}}^v\\ \mathbf{u}_1^v\\ \mathbf{u}_2^v\end{bmatrix}=\begin{bmatrix}\mathbf{G}^s & \bar{\mathbf{G}}^s & \mathbf{G}^v & \bar{\mathbf{G}}^v\end{bmatrix}\begin{bmatrix}\mathbf{q}^s\\ \bar{\mathbf{q}}^s\\ \mathbf{q}^v\\ \bar{\mathbf{q}}^v\end{bmatrix}, \tag{23}$$

with

$$\bar{\mathbf{u}}^s=\begin{pmatrix}\bar{u}_1^s\\ \bar{u}_2^s\\ \vdots\\ \bar{u}_n^s\end{pmatrix},\mathbf{u}^s=\begin{pmatrix}u_{n+1}^s\\ u_{n+2}^s\\ \vdots\\ u_N^s\end{pmatrix},\bar{\mathbf{u}}^v=\begin{pmatrix}\bar{u}_{N+1}^v\\ \bar{u}_{N+2}^v\\ \vdots\\ \bar{u}_{N+m}^v\end{pmatrix},\mathbf{u}_1^v=\begin{pmatrix}u_{N+m+1}^v\\ u_{N+m+2}^v\\ \vdots\\ u_{N+m+m_1}^v\end{pmatrix},\mathbf{u}_2^v=\begin{pmatrix}u_{N+m+m_1+1}^v\\ u_{N+m+m_1+2}^v\\ \vdots\\ u_M^v\end{pmatrix},$$

$$\mathbf{q}^s=\begin{pmatrix}q_1^s\\ q_2^s\\ \vdots\\ q_n^s\end{pmatrix},\bar{\mathbf{q}}^s=\begin{pmatrix}\bar{q}_{n+1}^s\\ \bar{q}_{n+2}^s\\ \vdots\\ \bar{q}_N^s\end{pmatrix},\mathbf{q}^v=\begin{pmatrix}q_{N+1}^v\\ q_{N+2}^v\\ \vdots\\ q_{N+m}^v\end{pmatrix},\bar{\mathbf{q}}^v=\begin{pmatrix}\bar{q}_{N+m+1}^v\\ \bar{q}_{N+m+2}^v\\ \vdots\\ \bar{q}_M^v\end{pmatrix},$$

where $\bar{\mathbf{u}}^s$, $\bar{\mathbf{q}}^s$ and $\mathbf{u}^s$, $\mathbf{q}^s$ are vectors containing known and unknown boundary variables of source nodes, respectively. $\bar{\mathbf{u}}^v$, $\bar{\mathbf{q}}^v$ and $\mathbf{u}_1^v$, $\mathbf{u}_2^v$, $\mathbf{q}^v$ represent vectors containing the known and unknown boundary variables of virtual nodes, respectively. $\bar{\mathbf{H}}^s$, $\mathbf{H}^s$, $\bar{\mathbf{H}}^v$, $\mathbf{H}_1^v$, and $\mathbf{H}_2^v$ and $\mathbf{G}^s$, $\bar{\mathbf{G}}^s$, $\mathbf{G}^v$, and $\bar{\mathbf{G}}^v$ are submatrices of **H** and **G**, respectively. The difference between $\mathbf{u}_1^v$ and $\mathbf{u}_2^v$ is that $\mathbf{u}_1^v$ represents a vector containing unknown potentials for virtual nodes at the corner, of which the adjacent edge is the Dirichlet boundary condition (e.g., node 22 in Fig. 3).

The boundary variables at virtual nodes in $\mathbf{u}_1^v$, $\mathbf{u}_2^v$, and $\mathbf{q}^v$ are not truly independent variables. To condense the degrees for these virtual nodes, additional constraint equations are provided. The potential in $\mathbf{u}_1^v$ is directly equal to the nodal value at the same geometric location on the adjacent edge (e.g., $u_{22}=\bar{u}_{35}$ in Fig. 3). In this case, $\mathbf{u}_1^v$ can be regarded as a vector in which all potentials are known.

When the virtual node is on the smooth part of the boundary (e.g., node 23 in Fig. 3), the potentials in $\mathbf{u}_2^v$ are approximated by Eq. (11). The virtual node is located at the sharper corner, whose adjacent edge is the Neumann boundary condition (e.g., node 25 in Fig. 3), the potential in $\mathbf{u}_2^v$ is calculated by Eq. (13). Then, $\mathbf{u}_2^v$ can be expressed as

$$\mathbf{u}_2^v=\mathbf{\Phi}_u\mathbf{u}^s, \tag{24}$$

where $\mathbf{\Phi}_u$ is the shape function matrix of the second layer interpolation for $\mathbf{u}^s$.

Using Eq. (12) for all normal fluxes in $\mathbf{q}^v$, then $\mathbf{q}^v$ is given by

$$\mathbf{q}^v = \mathbf{\Phi}_q \mathbf{q}^s, \tag{25}$$

where $\mathbf{\Phi}_q$ represents the shape function matrix of the second-layer interpolation for $\mathbf{q}^s$. Here, $\mathbf{\Phi}_u$ and $\mathbf{\Phi}_q$ are sparse matrices because the second-layer interpolation is a local approximation method.

Substituting Eqs. (24) and (25) into Eq. (23) and applying boundary conditions at each virtual node, the degrees for all virtual nodes in Eq. (23) are condensed

$$\begin{bmatrix} \bar{\mathbf{H}}^s & \mathbf{H}^s + \mathbf{H}_2^v \mathbf{\Phi}_u \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}}^s \\ \mathbf{u}^s \end{bmatrix} = \begin{bmatrix} \mathbf{G}^s + \mathbf{G}^v \mathbf{\Phi}_q & \bar{\mathbf{G}}^s \end{bmatrix} \begin{bmatrix} \mathbf{q}^s \\ \bar{\mathbf{q}}^s \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{G}}^v & -\bar{\mathbf{H}}^v & -\mathbf{H}_1^v \end{bmatrix} \begin{bmatrix} \bar{\mathbf{q}}^v \\ \bar{\mathbf{u}}^v \\ \mathbf{u}_1^v \end{bmatrix}, \tag{26}$$

By applying boundary conditions at all source nodes and switching the columns in the matrices, the final system of linear equations can be obtained

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{27}$$

with

$$\mathbf{A} = \begin{bmatrix} -(\mathbf{G}^s + \mathbf{G}^v \mathbf{\Phi}_q) & \mathbf{H}^s + \mathbf{H}_2^v \mathbf{\Phi}_u \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}^s \\ \mathbf{u}^s \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \bar{\mathbf{G}}^v & -\bar{\mathbf{H}}^v & -\mathbf{H}_1^v & \bar{\mathbf{G}}^s & -\bar{\mathbf{H}}^s \end{bmatrix} \begin{bmatrix} \bar{\mathbf{q}}^v \\ \bar{\mathbf{u}}^v \\ \mathbf{u}_1^v \\ \bar{\mathbf{q}}^s \\ \bar{\mathbf{u}}^s \end{bmatrix},$$

where **A** is a matrix of order *N*, **x** is a vector containing *N* boundary unknowns at source nodes only, and **b** is a known right-hand-side vector.

By means of the proposed method, the size of the overall system of linear equations is the same as that in the conventional discontinuous element implementation. The variables at virtual nodes do not occur in Eq. (27). This is a major advantage of our method. Solving Eq. (27) and combining Eqs. (24) and (25), we can obtain unknown nodal values for all interpolation nodes.

3.4 Algorithm of DLI-BEM

The algorithm of the DLI-BEM is as follows:

1. Discretize the boundary using DLIM elements.
2. Calculate the coefficient matrices **H** and **G** in Eq. (22).
3. Condense the degrees for all virtual nodes in Eq. (23) by using the boundary conditions and the additional constraint equations $\mathbf{u}_2^v = \mathbf{\Phi}_u \mathbf{u}^s$ and $\mathbf{q}^v = \mathbf{\Phi}_q \mathbf{q}^s$.
4. Impose the boundary conditions and solve the equation **Ax**=**b** by using LU decomposition so that potentials and normal fluxes at any source nodes are obtained.
5. Calculate the potentials and normal fluxes at virtual nodes by using the equations $\mathbf{u}_2^v = \mathbf{\Phi}_u \mathbf{u}^s$ and $\mathbf{q}^v = \mathbf{\Phi}_q \mathbf{q}^s$.

The potentials and normal fluxes at any boundary point are approximated by using the first-layer interpolation (e.g., Eqs. (11) and (12)).

4. Numerical examples

In this section, five numerical examples are presented. The first is used to test the interpolation accuracy of the DLIM. The other examples are given to demonstrate the accuracy and efficiency of the DLI-BEM for solving the Laplace equation,

$$\nabla^2 u = 0 \tag{28}$$

where ∇^2 is the Laplace operator. For purposes of error estimation and convergence study, a 'global' L_2 norm error, normalized by $|u|_{\max}$, is defined as

$$e = \frac{1}{|u|_{\max}} \sqrt{\frac{1}{M} \sum_{i=1}^{M} (u_i^{(e)} - u_i^{(n)})} \, , \tag{29}$$

where $|u|_{\max}$ is the maximum value over M sample nodes; the superscripts (e) and (n) refer to the exact and numerical solutions, respectively.

In all computations, unless indicated otherwise, the parameter d in Eqs. (2) and (3) is taken to be 0.25, and the support size of the weight function $\hat{d}_j$ in Eq. (10) is equal to 4.0h, 3.0h, and 1.5h for DLIM constant, linear, and quadratic elements, respectively. The h is the maximum size of elements on each boundary edge. The parameter c_j is taken to be such that $\hat{d}_j / c_j$ is a constant and equal to 3.5.

In the following figures, n is the total number of source nodes. DLI-BEM Const, DLI-BEM Linear, and DLI-BEM Quad are the numerical results of the DLI-BEM with constant, linear, and quadratic DLIM elements, respectively. Traditional BEM Linear, traditional BEM Quad, and traditional BEM Cubic represent the numerical results of the traditional BEM with the conventional continuous linear, quadratic, and cubic elements, respectively. The numerical results of the flux inside the domain are evaluated by

$$q = \sqrt{q_x^2 + q_y^2} \,, \tag{30}$$

where q_x and q_y are the component of q in the x and y directions, respectively.

4.1 Example 1

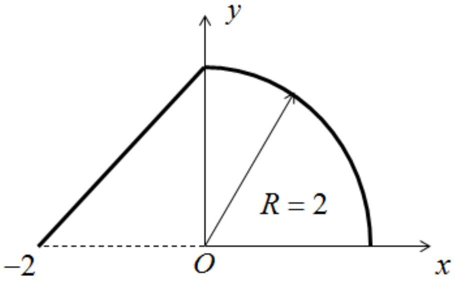


Fig. 5. Interpolation test on a straight line and a quadrant.

The first example is an interpolation test for a piecewise function, the model of which consists of a straight line (from (−2, 0) to (0, 2)) and a quadrant of radius 2 units centered at the origin (Fig. 5). The piecewise function $f(x, y)$ with a jump discontinuity at point (0, 2) is given by

$$f(x,y) = \begin{cases} -\exp(x^4 y^4 - x^3 y^3 + x^2 y^2 - xy - 1), & x \in [-2,0) \\ \exp(-x^3 + y^3 + x^2 - y^2 + xy - 2)\sin(x-y)^2, & x \in [0,2] \end{cases} . \tag{31}$$

To compare the interpolation accuracy between the DLIM and conventional discontinuous elements interpolation, the model is meshed by these two types of elements. The relative errors on each curve, with 12, 24, 48, 96, and 204 source nodes, are shown in Fig. 6. Numerical results on the quadrant with 36 source nodes, together with the exact solution, are shown in Fig. 7. In Figures 6 and 7, DLIM-Const, DLIM-Linear, DLIM-Quad, Discontinuous-Linear, Discontinuous-Quad, and Discontinuous-Cubic represent the results of DLIM and the discontinuous elements interpolation with different element orders.

Figure 6 shows that the DLIM has higher interpolation accuracy and convergence rates. Additionally, as plotted in Fig. 7, the method guarantees continuity on a smooth boundary and accurately approximates the discontinuous boundary variables at point (0, 2).

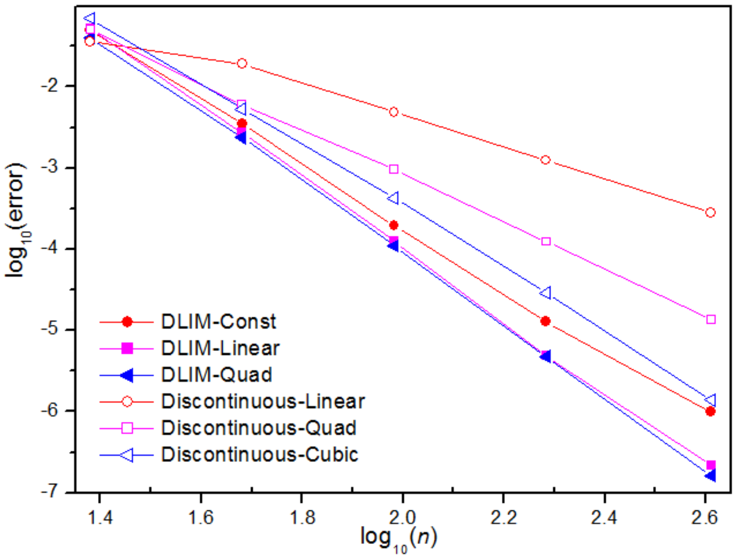


Fig. 6. Relative errors of $f(x,y)$ on the quadrant and straight line

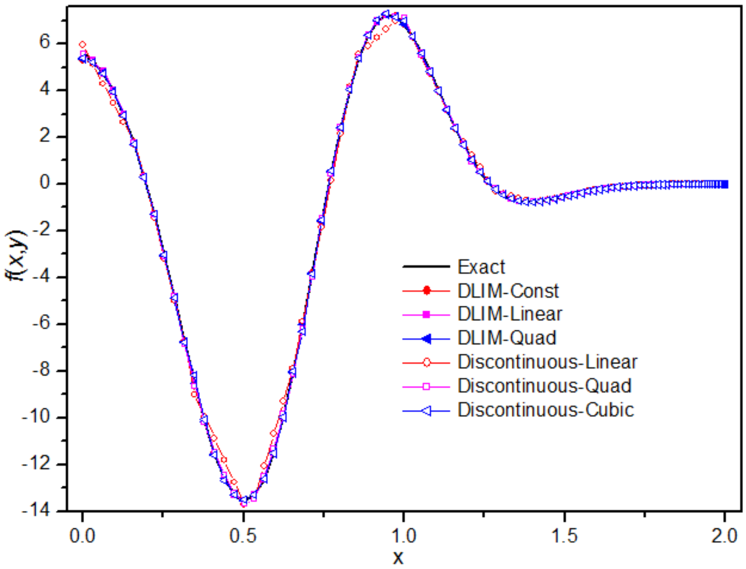


Fig. 7. Function values of $f(x,y)$ along the quadrant.

4.2 Example 2

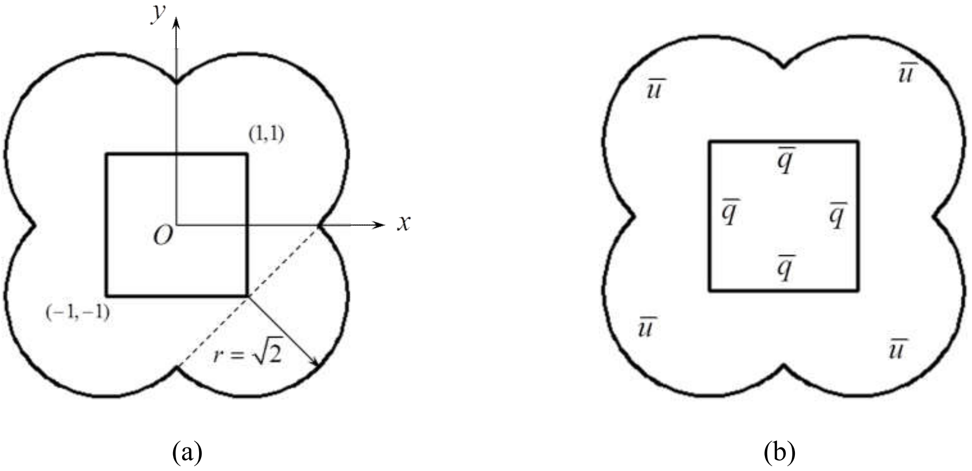


Fig. 8. Mixed problem on complicated geometry: (a) geometric model and (b) boundary conditions.

The case of a mixed problem on a complicated geometry is considered as the second example, and the size of the complicated geometry is illustrated in Fig. 8 (a). The analytical solution for this problem is given by:

$$u(x,y)=-x^4-y^4+6x^2y^2 \tag{32}$$

The prescribed u and q values along all boundaries are shown in Fig. 8 (b). The Dirichlet boundary condition on all semicircles is imposed according to the above analytical solution. To study the convergence and efficiency of the proposed method, we employed five sets of source nodes on all straight lines, 6, 12, 18, 24 and 30. The number of source nodes on each semicircle is two times more than the nodes on any straight line.

Figs. 9 and 10 show the relative errors for u on Neumann boundary conditions and q on Dirichlet boundary conditions, respectively. Combined with the exact solution, the numerical results of u and q inside the domain along a semicircle (angle from π to 0) of radius 1.8 units, centered at the origin, are shown in Figs. 11 and 12, respectively. As illustrated in Figs. 9–12, high accuracy and convergence rates for the mixed problem with complicated geometry can be achieved with the method proposed.

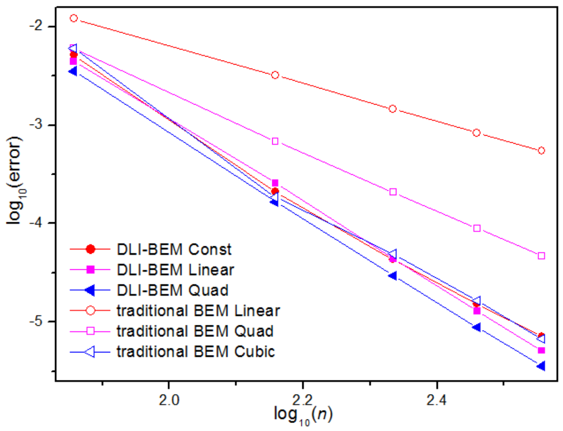


Fig. 9. Relative errors and convergence rate of *u* along all straight lines.

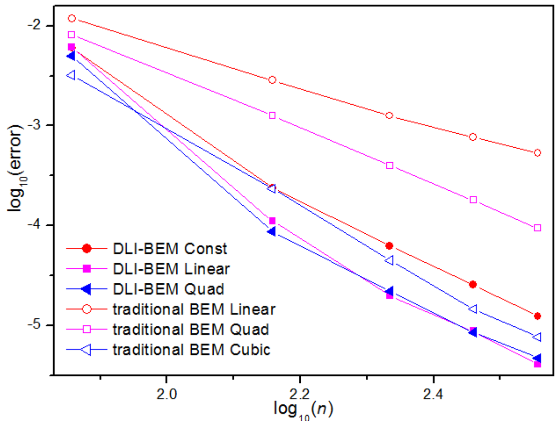


Fig. 10. Relative errors and convergence rate of *q* along all semicircles.

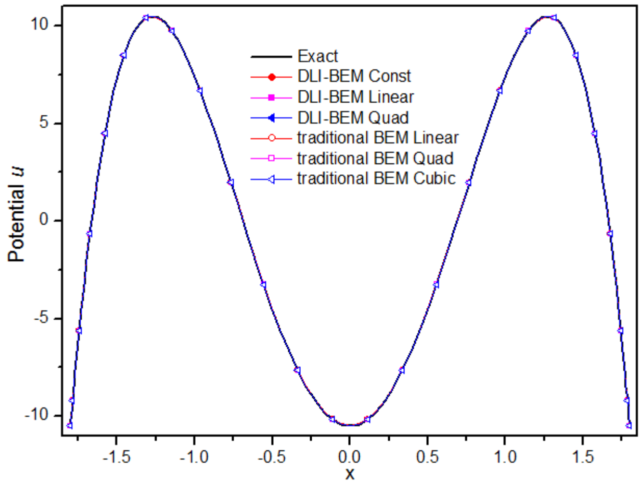


Fig. 11. Values of *u* along a semicircle of radius 1.8 units centered at the origin (with angle from π to 0).

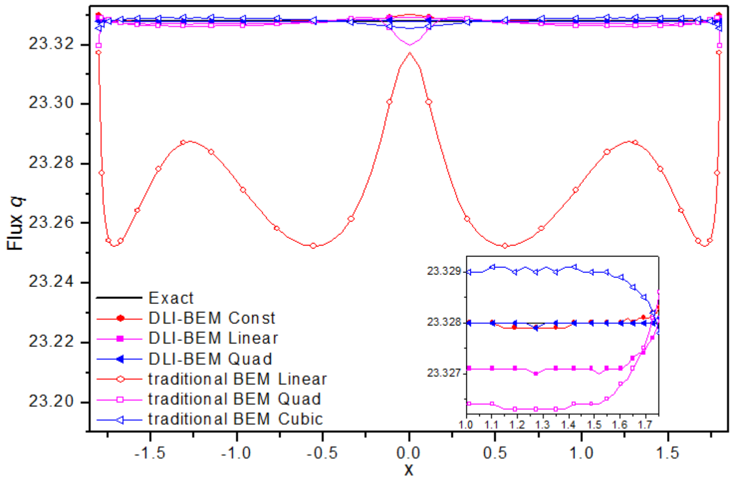


Fig. 12. Values of *q* along a semicircle of radius 1.8 units centered at the origin (with angle from π to 0).

4.2 Example 3

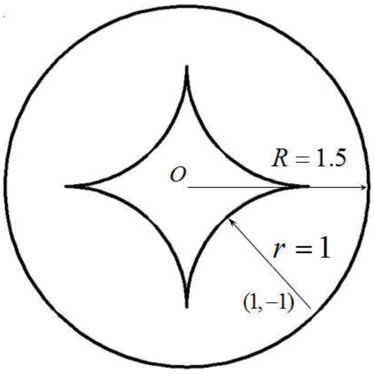


Fig. 13. Dirichlet problem with a complex geometry.

A circle of radius 1.5 units centered at the origin containing an asteroid is considered for the second example (Fig. 13). The analytical solution is given by

$$u(x, y) = -x^3 - y^3 + 3x^2y + 3xy^2 . \tag{32}$$

The Dirichlet boundary condition is imposed on all edges. In this example, there are four sets of source nodes on each interior edge, 6, 12, 24, and 48. Moreover, the number of source nodes on the exterior circle is six times more than the nodes on any interior edge.

The relative errors of q on all edges are plotted in Fig. 14. The numerical results of u and q along a semicircle (the angle from π to 0) of radius 1.48 units, centered at the origin, with six source nodes on each interior edge, are illustrated in Figs. 15 and 16, respectively. The results show that the proposed method has high rates of convergence. The agreement between numerical and exact results is excellent. The CPU time spent in constructing and solving the system equations is plotted in Fig. 17. It is clear that the proposed method requires less CPU time than the traditional BEM.

The effect of the parameter d in Eqs. (2) and (3) on the computational accuracy of the proposed method was studied in this example. The above four sets of source nodes are also used here, which yield 60, 120, 240, and 480 source nodes. As shown in Fig. 18, for the same number of source nodes, the accuracy of the DLI-BEM with DLIM linear element will be improved slightly with the increment of the offset of source node d, whereas the opposite phenomenon is seen for the DLI-BEM with the DLIM quadratic element. It can be inferred that the 'best' parameter d should be between 0.2 and 0.25.

To study the effect of the parameter c_j in Eq. (10) on the solution accuracy of the proposed method, we arranged 6 and 36 source nodes on all interior edges and the circle, respectively. As shown in Fig. 19, the results of the DLI-BEM with DLIM quadratic element are still stable with variation in $\hat{d}_j / c_j$, and the results of the DLI-BEM with DLIM constant and linear elements are also acceptable.

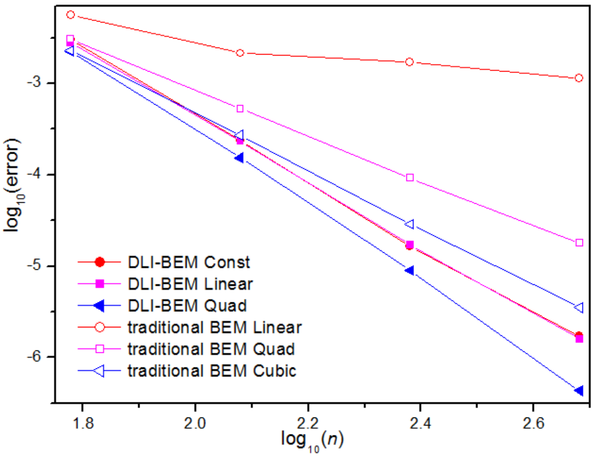


Fig. 14. Relative errors and convergence rates of q on all edges.

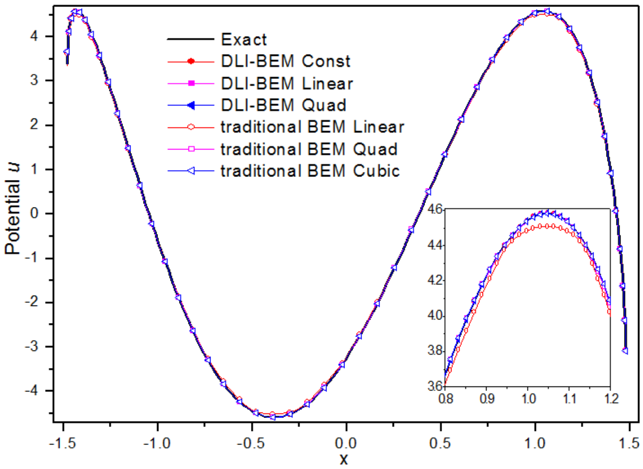


Fig. 15. Values of u along a semicircle of radius 1.48 units centered at the origin (with the angle from π to 0).

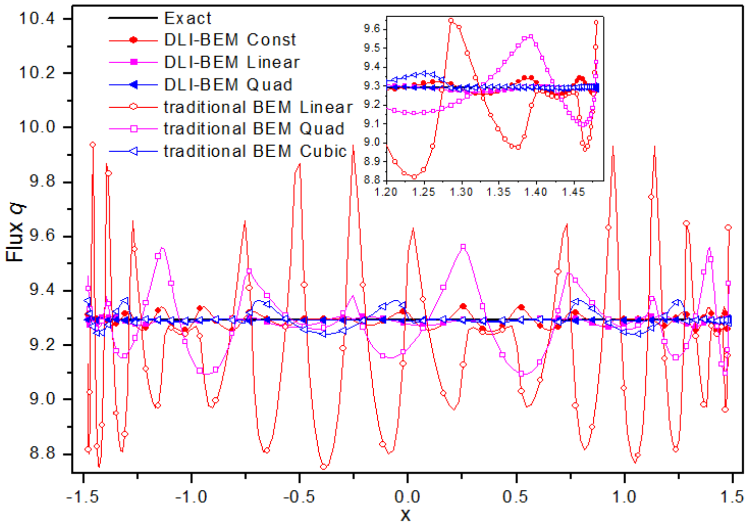


Fig. 16. Values of q along a semicircle of radius 1.48 units centered at the origin (with the angle from π to 0).

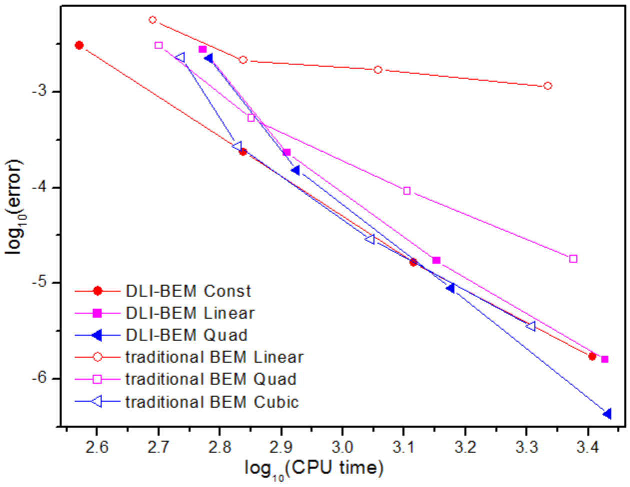


Fig. 17. Comparison of computational efficiency for a Dirichlet problem.

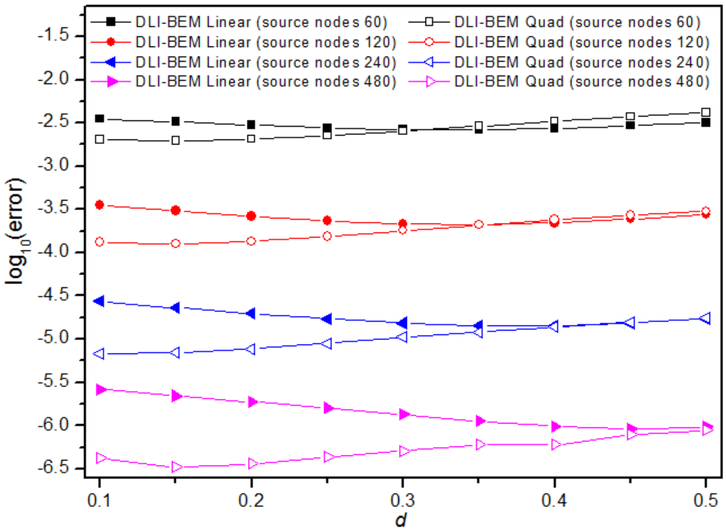


Fig. 18. Effect of d on the accuracy of the DLI-BEM for the Dirichlet problem.

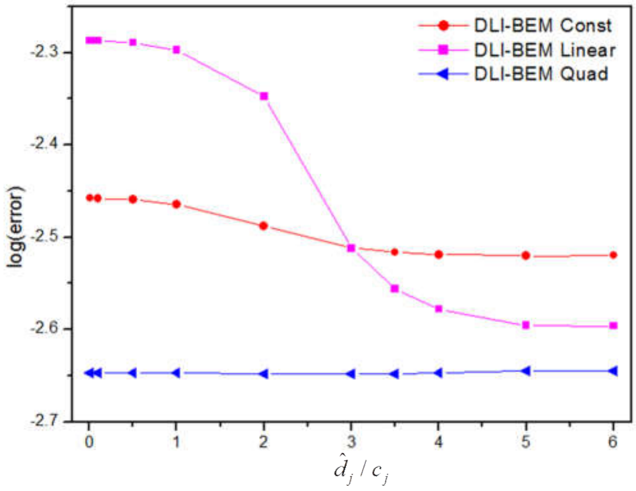
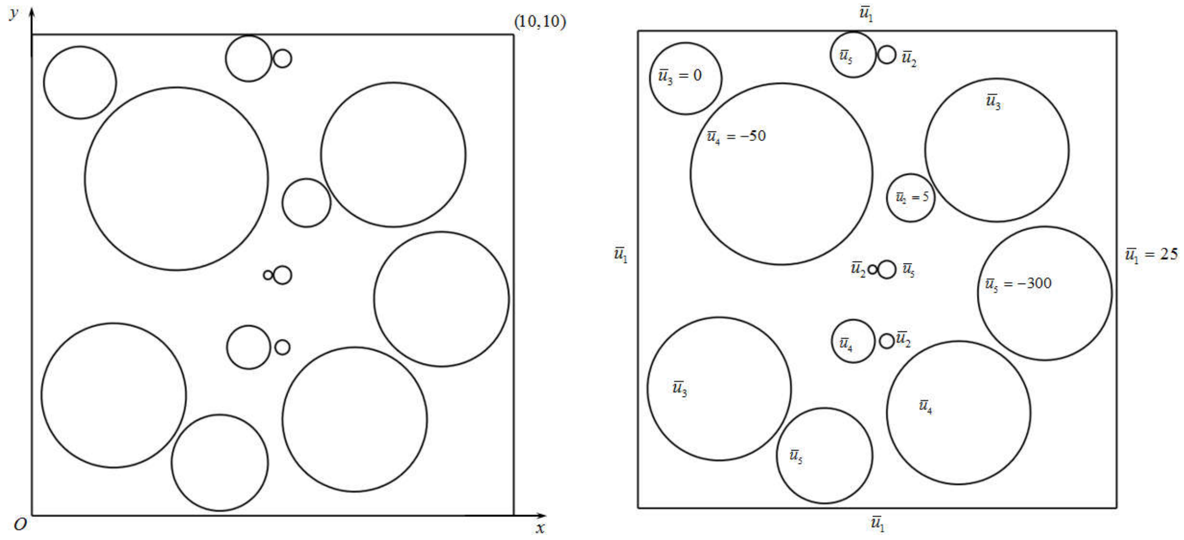


Fig. 19. Effect of $\hat{d}_j / c_j$ on the accuracy of the DLI-BEM for approximating u.

4.4 Example 4



(a) (b)

Fig. 20. Mixed problem on a square domain with internal cavities: (a) geometric model and (b) boundary conditions.

The fourth example is presented to show the accuracy and convergence of the DLI-BEM with comparison to the FEM. In this example, the geometric model is constituted by a 10×10 square domain with many internal cavities (see Fig. 20 (a)), and the prescribed u along all boundaries are shown in Fig. 20 (b).

In DLI-BEM the potentials and normal fluxes on the boundary are approximated by quadratic DLIM elements, while these physical variables are approximated by quadratic triangle elements in FEM. Table 1 lists the numbers of elements and source nodes for the proposed method and FEM. In the following figures of this example, the results of DLI-BEM270 and DLI-BEM462 are obtained by the proposed method using 270 and 432 source nodes, respectively. The results of FEM with 1,802 and 4,822 source nodes are denoted by FEM1802 and FEM4822, respectively. The numerical results by the FEM with 563,037 sources nodes are used as a reference solution.

The numerical results of the potential along a line (x=5) are shown in Fig. 21, and the potential distribution in the whole domain is shown in Fig. 22. From Fig. 21, one can see that the numerical results of the potential by both methods approach to the reference solution as the number of node increases, but the convergence rate by the proposed method are higher than that by the FEM. As shown in Fig. 22, high level of accuracy can be obtained by the proposed method with few source nodes.

Table 1 Number of elements and source nodes in example 4.

	DLI-BEM		FEM		
	1	2	1	2	3
Elements	90	154	779	2183	278706
Source nodes	270	462	1802	4822	563037

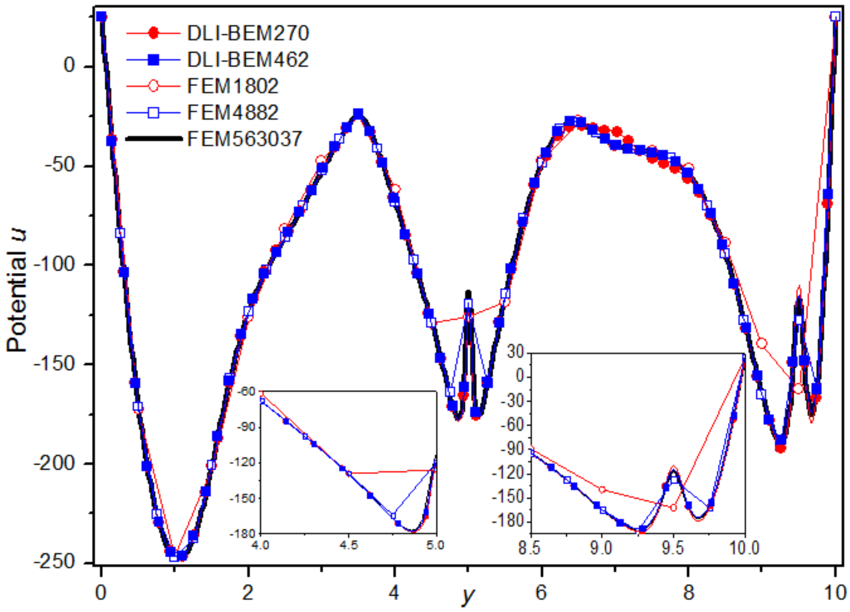
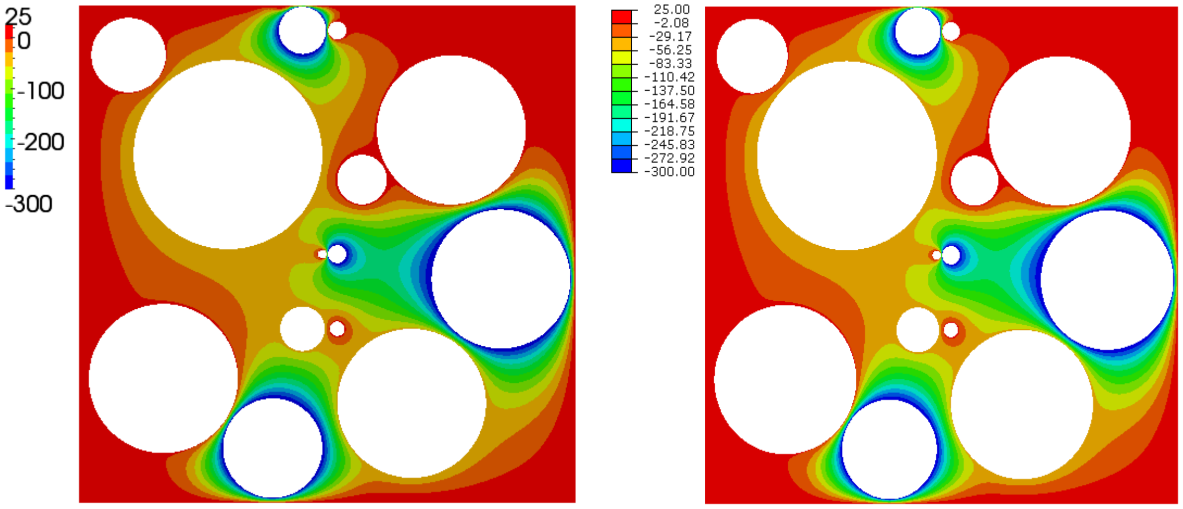


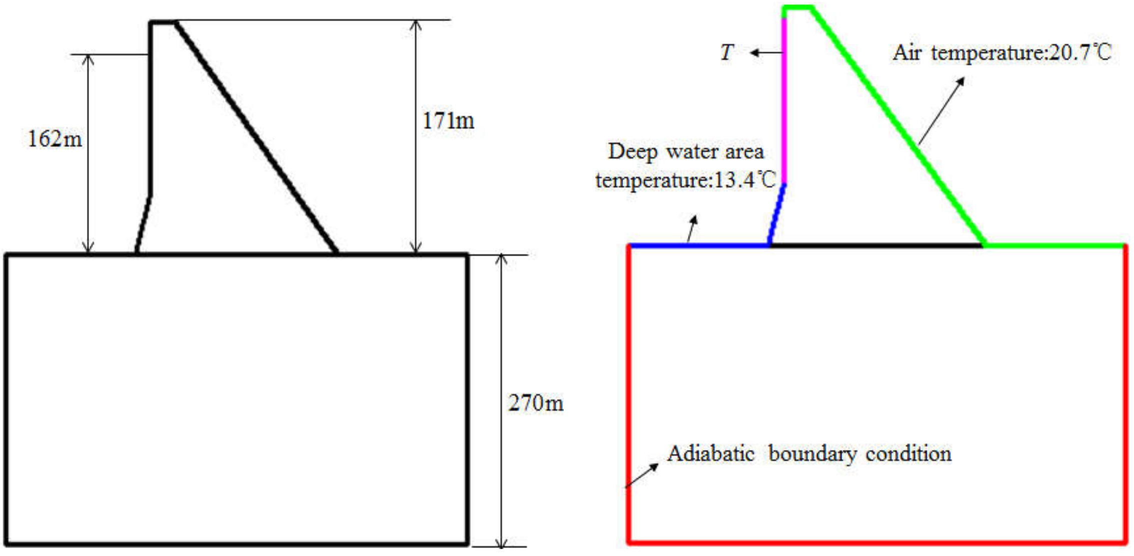
Fig. 21. Values of potential *u* along a line x=5.



(a) (b)

Fig. 22. Potential distribution by: (a) DLI-BEM with 462 source nodes and (b) FEM with 563037 source nodes.

4.5 Example 5



(a) (b)

Fig. 23. Steady heat conduction problem on a dam: (a) geometric model and (b) boundary conditions.

A steady heat conduction problem on a real dam model (with the coefficient of heat conduction $k = 1$) is considered in the last example (Fig. 23). The interface is 157 m between the dam and the bed rock. The left and right of the dam are regarded as upstream and downstream, respectively. The upstream water level is 162 m and the distance is 9 m between it and the top of the dam. The other edges, however, are exposed to air.

The left, right, and bottom edges of the bed rock are set as adiabatic boundaries and the annual average air temperature of the location is taken as 20.7°C (Fig. 23(b)). Moreover, the temperature of water at a depth more than 114 m is taken as 13.4°C, while the temperature of water within 114 m is given by

$$T = 20.7 - 0.0640350877 \times h , \tag{34}$$

where h is the depth into the water.

For the DLI-BEM, 6 source nodes are arranged on each edge, yielding a total number of 60 source nodes. However, two different nodes arrangements of 6 and 18 nodes on each edge are used in FEM, yielding 152 and 1,295 source nodes, respectively. In the following figures of this example, the results of FEM152 and FEM1295 are obtained by FEM, corresponding to 152 and 1,295 nodes, respectively. The result obtained by FEM with 137,354 source nodes is considered a reference solution, denoted as RefSolution. The temperature along the interface between dam and bed rock is shown in Fig. 24, and the temperature distribution in the domain is shown in Fig. 25.

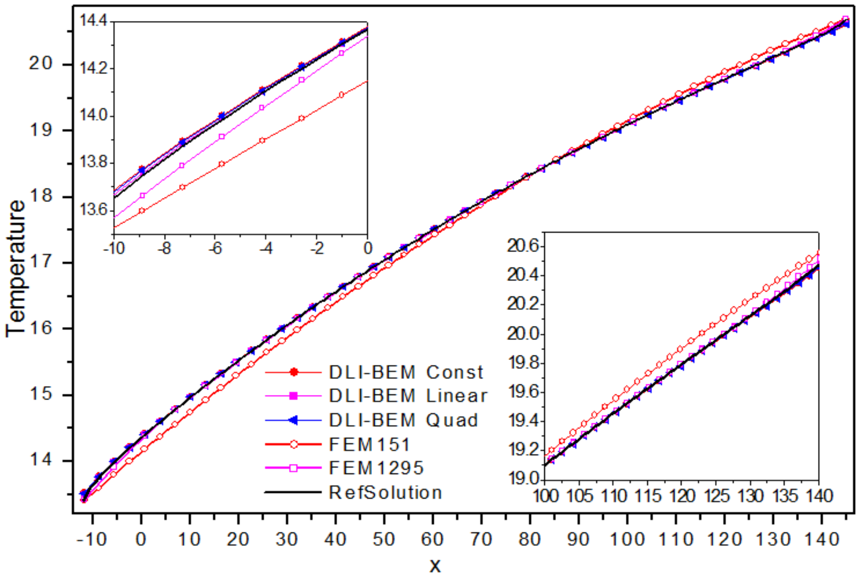


Fig. 24. Temperature along the interface between the dam and the bed rock.

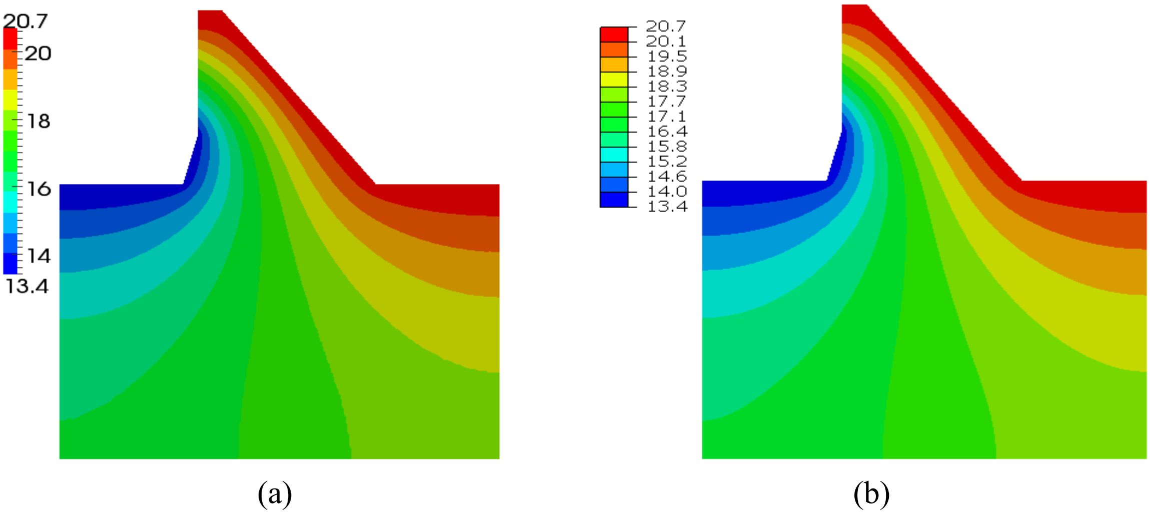


Fig. 25. Temperature distribution: (a) DLI-BEM Const and (b) RefSolution.

5. Conclusions and discussions

In this paper, we have proposed a double-layer interpolation method (DLIM) by combining the conventional polynomial element interpolation and the MLS approximation. The new method inherits the advantages from both the element interpolation and the element-free approximation, while overcoming their disadvantages. Compared with the continuous and discontinuous element interpolation methods, the DLIM is able to naturally and accurately approximate both continuous and discontinuous fields. Thus, the DLIM has unified the continuous and discontinuous element interpolations. Numerical tests have demonstrated that it possesses a remarkable accuracy and high rates of convergence.

The DLIM provides a new implementation for the BEM. In the DLI-BEM, the MLS is not used to evaluate the shape functions at each Gaussian point in the numerical integration, but rather only to construct the relationship between the source nodes and the virtual nodes. Furthermore, source nodes covered in the influence domain of a virtual node can be obtained directly by searching neighboring elements of the corresponding virtual node. Thus, the computational efficiency of the MLS in the DLI-BEM is much higher than that of MLS constructed by scattered nodes in a pure element-free method. In the DLI-BEM, the computational cost of the numerical integration and the matrix assembly is slightly increased. However, for the same level of accuracy, the DLI-BEM requires less computational cost. Moreover, it simplifies the evaluation of hypersingular integrals, and the treatment of geometric corners and discontinuous boundary conditions.

Our method was confirmed through a number of numerical examples with different geometries and boundary condition types. Compared with the traditional BEM, the DLI-BEM possesses much higher accuracy, convergence rates and computational efficiency for most cases. It is demonstrated that the DLIM can substantially improve the performance of the BEM. Due to the above advantages, the new method is promising for solving problems involving discontinuity, such as crack propagation, contact problems, etc.

Acknowledgements

This work was supported by National Science Foundation of China under grant number 11472102.

References

[1] M.A. Jaswon, Integral equation methods in potential theory I, Proc R Soc Lond, A275 (1963), 23–32.

[2] G.T. Symm, Integral equation methods in potential theory II, Proc R Soc Lond, A275 (1963), 33–46.

[3] J.M. Zhang, X.Y. Qin, X. Han, G.Y. Li, A boundary face method for potential problems in three dimensions, Internat. J. Numer. Methods Engrg. 80 (3) (2009) 320-337.

[4] L. Sun, W. Chen, C. Zhang. A new formulation of regularized meshless method applied to interior and exterior anisotropic potential problems. Applied Mathematical Modelling, 2013, 37(12): 7452-7464.

[5] W. Qu, W. Chen, Z. Fu, Solutions of 2D and 3D non-homogeneous potential problems by using a boundary element-collocation method, Eng. Anal. Bound. Elem. 60 (2015): 2-9.

[6] Y.M. Zhang, F.L. Sun, D. Young, Average source boundary node method for potential problems, Eng. Anal. Bound. Elem. 70 (2016): 114-125.

[7] C.A. Brebbia, S. Walker, Boundary element techniques in engineering, Elsevier, 2016.

[8] S. Kobayashi, N. Nishimura, Boundary Element Methods: Fundamentals and Applications, Springer Science & Business Media, 2013.

[9] A.H.D. Cheng, D.T. Cheng, Heritage and early history of the boundary element method, Eng. Anal. Bound. Elem. 29 (3) (2005) 268–302.

[10] F.L. Zhou, Y. Li, J.M. Zhang, C. Huang, C.J. Lu, A time step amplification method in boundary face method for transient heat conduction, Int. J. Heat Mass Tran. 84 (2015) 671-679.

[11] J.M. Zhang, M. Tanaka, T. Matsumoto, A simplified approach for heat conduction analysis of CNT-based nano-composites, Comput. Methods Appl. Mech. Engrg. 193 (2004) 5597-5609.

[12] M.J. Peake, J. Trevelyan, G. Coates, Extended isogeometric boundary element method (XIBEM) for three-dimensional medium-wave acoustic scattering problems, Comput. Methods Appl. Mech. Engrg. 284 (2015) 762–780.

[13] X.H. Wang, J.M. Zhang, F.L. Zhou, X.S. Zheng, An adaptive fast multipole boundary face method with higher order elements for acoustic problems in three-dimension, Eng. Anal. Bound. Elem. 37 (2013) 114–152.

[14] Y.M. Zhang, V. Sladek, J. Sladek. A new boundary integral equation formulation for plane orthotropic elastic media. Applied Mathematical Modelling, 36.10 (2012): 4862-4875.

[15] G.Z. Xie, J.M. Zhang, C. Huang, C.J. Lu, G.Y. Li, A direct traction boundary integral equation method for three-dimension crack problems in infinite and finite domains, Comput. Mech. 53 (4) (2014) 575-586.

[16] G.D. Manolis, P.K. Banerjee, Conforming versus non-conforming boundary elements in three-dimensional elastostatics, Internat. J. Numer. Methods Engrg, 23 (1986) 1885-1904.

[17] P. Parreira, On the accuracy of continuous and discontinuous boundary elements, Engineering Analysis, 5 (4) (1988) 205-211

[18] M. Guiggiani, G. Krishnasamy, T.J. Rudolphi, F.J. Rizzo, A general algorithm for the numerical solution of hypersingular boundary integral equations, J. Appl. Mech. 59 (1992) 604-614.

[19] G. Krishnasamy, F.J. Rizzo, T.J. Rudolphi, Continuity requirements for density functions in the boundary integral equation method, Comput. Mech. 9 (1992) 267-284.

[20] S. Mukhopadhyay, N. Majumdar, A study of three-dimensional edge and corner problems using the neBEM solver, Eng. Anal. Bound. Elem. 33 (2009) 105-119.

[21] X.S. Zheng, J.M. Zhang, K. X.M. Shu, L. Han, Boundary face method for 3D contact problems with non-conforming contact discretization, Eng. Anal. Bound. Elem. 63 (2016) 40-48.

[22] X.M. Shu, J.M. Zhang, L. Han, Y.Q. Dong, A surface-to-surface scheme for 3D contact problems by boundary face method, Eng. Anal. Bound. Elem. 70 (2016) 23-30.

[23] P. Lancaster, K. Salkauskas, Surface generated by moving least squares methods, Math. Comput. 37 (1981) 141–158.

[24] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Element-free Galerkin methods, Int. J. Numer. Methods Engrg. 37 (1994) 229-256.

[25] J. Wang, F. Sun, Y. Cheng, An improved interpolating element-free Galerkin method with nonsingular weight function for two-dimensional potential problems, Chin. Phys. B 21 (9) (2012) 090204.

[26] X.L. Li. An interpolating boundary element-free method for three-dimensional potential problems. Applied Mathematical Modelling, 2015, 39(10): 3116-3134.