Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 390 (2022) 114432
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Dual interpolation boundary face method for 3-D potential problem based on binary tree grids

Jianming Zhang[a,*], Rongxiong Xiao[a], Pihua Wen[b], Chuanming Ju[c], WeiCheng Lin[a], Rui He[a]

[a] *State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China*
[b] *School of Engineering and Materials Science, Queen Mary University of London, London EI 4NS, UK*
[c] *Yantai University, Yantai 264005, China*

Received 27 June 2021; received in revised form 22 October 2021; accepted 5 December 2021
Available online 6 January 2022

Abstract

This paper presents the dual interpolation boundary face method (DiBFM) based on a binary tree grid for solving the 3-D potential problems. The subdivision method based on binary tree algorithm is capable to generate both continuous and discontinuous grids, and to achieve the grids generation for arbitrary complex model much easier and automatically. By adding virtual nodes on the vertices and edges of traditional discontinuous element, the dual interpolation elements are introduced while the order is increased by two. The values of physical variables are approximated by Lagrange interpolation polynomial in the first-layer interpolation, and meshless interpolation is used to condense the degree of freedom of virtual nodes in the second-layer interpolation. Since no requirement is needed for the continuity of grids in the DiBFM, the dilemma of discontinuous grids are avoidable. In this paper, the DiBFM is implemented based on the binary tree grids to solve general problems with thin-wall structures and "geometrical construction noise".

Keywords: DiBFM; Discontinuous grids; Meshless interpolation

1. Introduction

The boundary element method (BEM) based on the boundary integral equation (BIE) has been developed rapidly [1–3] in recent decades and widely applied to different fields including acoustics [4], fracture mechanics [5], elastoplasticity [6], potential [7]. Different from the finite element method (FEM), there is no requirement of trial function and only discretization on the boundary by using BEM. Hence, boundary discontinuous elements can also be used in BEM rather than domain elements in FEM.

With the wide application of CAE software in modern industrial manufacturing, it is of great significance to realize the automation of CAE analysis and the integration of CAE/CAD, and one key is the automatic computational grids generation. In the implementation of FEM, the grids need to be continuous, and the computation results are

* Correspondence to: College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China.
E-mail address: zhangjm@hnu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.cma.2021.114432
0045-7825/

quite dependent on the quality of grids, which requires the simplification on the CAD model with small features such as "small chamfers and notches". On the one hand, simplification automatically is difficult. On the other hand, the correctness of the calculation results cannot be guaranteed since singularities may appear in the simplified model. In order to overcome the problems caused by the independence of CAE and CAD models, T.J.R. Hughes proposed an isogeometric analysis (IGA) method [8], which uses the same NURBS function to express the geometric structure and approximate physical field. This method unified CAD and CAE models, and becomes a major research hotspot [9,10]. In the early research, the IGA was mostly applied to the finite element method. Subsequently, the isogeometric BEM (IGABEM) which coupled the IGA with the BEM was developed and has been applied to many fields [11,12]. However, the NURBS model in IGA inevitably has gaps or overlaps at the surface intersection, which brings the problems of grid generation again. To overcome these difficulties, the IGA using T-splines was proposed [13]. But for arbitrary three-dimensional structure, it is a more challenging problem to convert the two-dimensional T-splines surface representation into a three-dimensional T-splines solid grid model. Furthermore, the IGA is not suitable for the structures with"geometric noise", where geometry restorations are required. Besides, the boundary face method (BFM) [14,15] is proposed to truly achieve the integration of CAD\CAE and CAE analysis is performed directly on CAD models. In the implementation of BFM, no processing of CAD models is needed, and the geometric errors are avoided since the integrals and physical variables are interpolated in the parameter space of surfaces for 3D problems.

In the implementation of BFM and conventional BEM, the choice of continuous element or discontinuous element has always been controversial [16] and either has its pros and cons. When using discontinuous elements, it is convenient to deal with singular integrals and can reduce the difficulty of grid generation, but the interpolation accuracy is poorer than that when using continuous elements with the same number of source points. While using continuous element, the C^0 continuity of variables can be ensured, but it is troublesome to handle the corner problem. To unify continuous and discontinuous elements, the DiBFM [17] is developed by implementing the BFM based on the dual interpolation element. These elements include two kinds of node: source and virtual nodes. The source nodes are arranged inside in the elements and the virtual are put on the vertices and edges in 3-D problem. The values of variables are approximated by Lagrange interpolation polynomial in the first-layer interpolation. Since only the source nodes are the collocation points of BIE, in the implementation of the second-layer interpolation, the meshless interpolation is constructed to interpolate variable values of the virtual nodes with the source. The advantage that trial function does not be continuous is fully utilized in DiBFM and the accuracy gets improved. What is more, the grid obtained by the binary tree subdivision method (BTSM) [18] can be applied to the implementation of DiBFM since there is no requirement of continuous grids.

Compared with the traditional continuous grid generation methods, such as the advancing front method [19], Delaunay triangulation method [20], mapping method [21], BTSM can generate both continuous and discontinuous grids, and make the grids generation automatically for arbitrary complex models.

In this paper, the Hermite-type moving-least-squares (HMLS) approximation proposed by Li and Aluru [22,23] is changed with the standard moving-least-squares (MLS) [24,25] interpolation in the implementation of the DiBFM. In the implementation of MLS interpolation, only the unknown potential need to be considered in the weighted expression and the shape functions are constructed in the cylindrical coordinates systems for the boundary value problem. However, in the HMLS interpolation, the normal flux is also taken into account and the shape functions are constructed in Cartesian coordinates so that the influence domain of the weight function has more than one surface. Therefore, the DiBFM with HMLS interpolation is more advanced and efficient for solving practical problems with thin-wall structures and geometrical construction noise.

Section 2 introduces the binary tree grids. The dual interpolation method in 3-D problem is introduced in Section 3. Section 4 demonstrates the DiBFM with HMLS interpolation. Numerical examples are given in Section 5. Section 6 drafts the conclusions.

2. Binary tree grids

The grids obtained in the binary tree subdivision method are called binary tree grids. These grids are continuous and discontinuous grids as shown in Fig. 1. Compared with the traditional mesh generation methods with the continuous grids, this method can use hanging points, and is more suitable to mesh models with complex geometric configuration. In addition, this method is able to control the mesh density flexibly and to generate anisotropic grids more naturally.

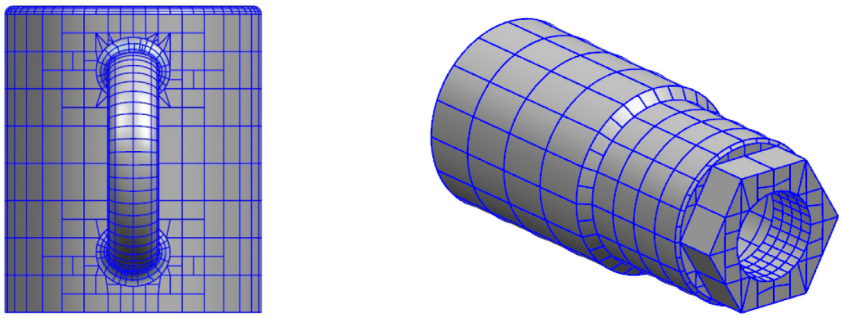


Fig. 1. Binary tree grids.

The fan blade is used as example to show the implementation process of the binary tree subdivision method as depicted in Fig. 2

The binary tree mesh generation process of the surface is implemented in the parameter space of the surface, and the algorithm is described as

① First, determine the initial root element which is defined by the largest rectangular bounding area in the domain of the surface.

② Secondly, subdivide according to the given mesh size and the curvature of the surface.

③ Then subdivide the boundary curve of the surface according to the boundary conditions, and obtain the discrete points on the boundary.

④ The leaf elements are further divided according to the distribution of discrete points, so that the sizes of adjacent leaf elements are well transitioned.

⑤ Find the intersection of the boundary curve and the edge of the leaf element, and judge the position for all nodes from the intersection on the edge of the element.

⑥ The nodes close to the boundary curve are moved to the boundary curve.

⑦ Finally, after template processing, i.e. eliminating the short edges and large angles, the final mesh generation result is generated.

It is worth noting that, combined with the DiBFM, the binary tree grids can heritage all advantages of the BIE without the requirement of the trial function continuity. One key of the integration of CAE/CAD is the automatic computational grid generation and the discontinuous grid generation method based on the binary tree algorithm makes the automatic grid generation with arbitrary complex structures feasible.

3. Dual interpolation method

In the implementation of this method, there are two interpolations and the dual interpolation element is introduced. These elements include two types of nodes: source and virtual nodes. The source nodes locate inside the elements while the virtual nodes on the edges and the vertices as shown in Fig. 3.

Fig. 4 shows an example of virtual node layout. In general, virtual nodes are assigned according to the following rules:

(1) The virtual nodes of the same geometric surface or smooth geometric boundary at the same mesh vertex are shared, as shown by the point Q_1^v;
(2) The virtual nodes of the same geometric surface or smooth geometric boundary at the same grid edge are shared, as shown by the point Q_2^v;
(3) The virtual nodes at the vertices of the mesh on the non-smooth geometric boundary are generated separately, as shown by the point Q_3^v;
(4) The virtual nodes at the edges of the mesh on the non-smooth geometric boundary are generated separately, as shown by the point Q_4^v.

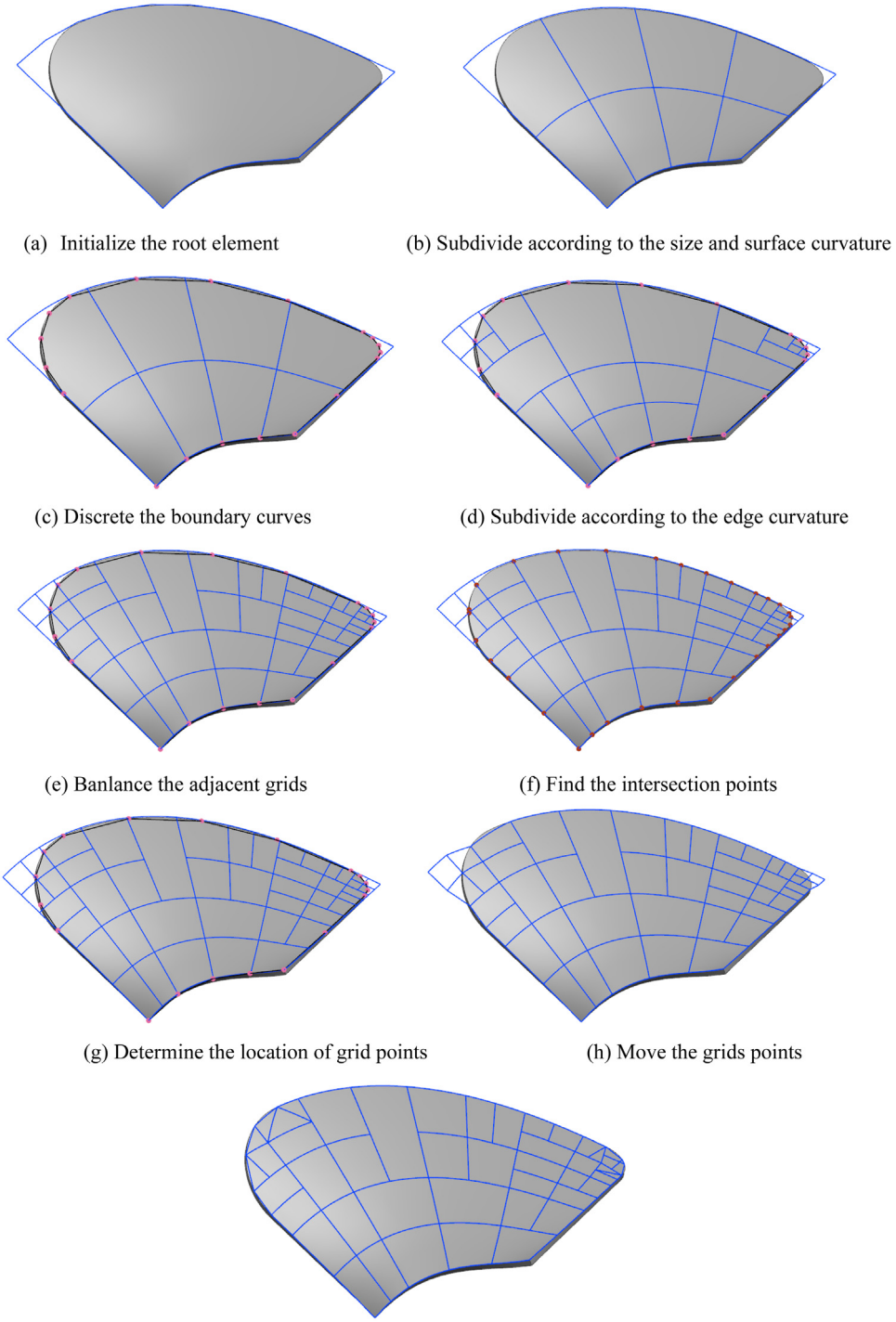


Fig. 2. The implementation process of the BTSM.

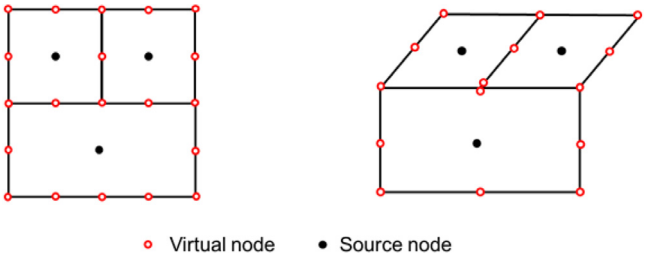


Fig. 3. Dual interpolation constant elements based on binary tree grids.

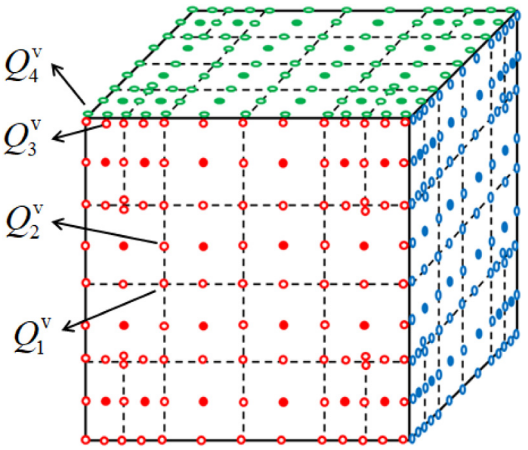


Fig. 4. The layout scheme of virtual nodes.

3.1. First-layer interpolation

The boundary physical variables are approximated with Lagrange interpolation polynomial in this interpolation. Different from the interpolation of traditional discontinuous element, the contribution of virtual nodes also needs to be taken into account. For the potential u and its normal derivative q in 3-D potential problem, the first-layer interpolation form [17] is

$$u(\xi,\eta)=\sum_{\alpha=1}^{n_\alpha}N_\alpha^s(\xi,\eta)u(Q_\alpha^s)+\sum_{\beta=1}^{n_\beta}N_\beta^v(\xi,\eta)u(Q_\beta^v), \tag{1}$$

$$q(\xi,\eta)=\sum_{\alpha=1}^{n_\alpha}N_\alpha^s(\xi,\eta)q(Q_\alpha^s)+\sum_{\beta=1}^{n_\beta}N_\beta^v(\xi,\eta)q(Q_\beta^v), \tag{2}$$

where ξ and η are the coordinates, the superscripts s indicate the source node and v virtual node, n_α and n_β are the number of source and virtual nodes, $u(Q_\alpha^s)$, $q(Q_\alpha^s)$, and $N_\alpha^s(\xi,\eta)$ represent the potential, normal flux, and shape function of the source node, respectively, and $u(Q_\beta^v)$, $q(Q_\beta^v)$, and $N_\beta^v(\xi,\beta)$ are the similar variables corresponding to the virtual node, respectively.

3.2. Second-layer interpolation with HMLS interpolation

Since the virtual nodal values in the first-layer interpolation are not independent, the meshless interpolation is used to approximate the variable values of virtual nodes with the source in this interpolation. The detail of HMLS interpolation is described below

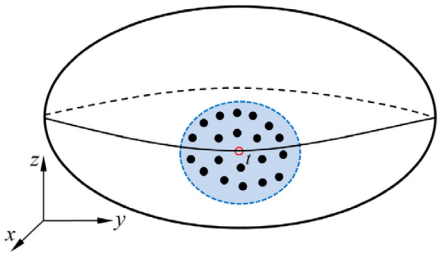


Fig. 5. Point t and its neighborhood.

For a point t given in Fig. 5, the potential u and normal flux q in its neighborhood are given as:

$$u(x, y, z) = \mathbf{p}_u^T(x, y, z)\mathbf{a}_t, \tag{3}$$

$$q(x, y, z, n) = \mathbf{p}_q^T(x, y, z, n)\mathbf{a}_t, \tag{4}$$

where $\mathbf{p}_u(x, y, z)$ and $\mathbf{p}_q(x, y, z, n)$ are the basis functions of potential u and its normal derivative q, respectively, superscript T stands for transpose. $\mathbf{a}_t$ denotes the unknown coefficient, (x, y, z) are the Cartesian coordinates and n represents the outward normal.

To obtain $\mathbf{a}_t$, the residual J is defined as:

$$J = \sum_{k=1}^{M} w_k[\mathbf{p}_u^T(x_k, y_k, z_k)\mathbf{a}_t - u_k]^2 + \sum_{k=1}^{M} w_k[\mathbf{p}_q^T(x_k, y_k, z_k, n_k)\mathbf{a}_t - q_k]^2, \tag{5}$$

where M is the number of interpolation points in the neighborhood of point t, w_k, u_k and q_k are the weighted function, potential and normal flux of interpolation point k, respectively.

Setting $\frac{\partial J}{\partial \mathbf{a}_t} = 0$ and $\mathbf{a}_t$ is derived as:

$$\mathbf{a}_t = \mathbf{C}_t^{-1}(\mathbf{P}_u^T\mathbf{W}\mathbf{u} + \mathbf{P}_q^T\mathbf{W}\mathbf{q}), \tag{6}$$

where

$$\mathbf{C}_t = \mathbf{P}_u^T\mathbf{W}\mathbf{P}_u + \mathbf{P}_q^T\mathbf{W}\mathbf{P}_q, \tag{7}$$

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_M \end{bmatrix}, \tag{8}$$

$$\mathbf{P}_u^{\mathbf{T}} = \begin{bmatrix} \mathbf{p}_u(x_1, y_1, z_1) & \mathbf{p}_u(x_2, y_2, z_2) & \cdots & \mathbf{p}_u(x_M, y_M, z_M) \end{bmatrix}, \tag{9}$$

$$\mathbf{P}_q^{\mathbf{T}} = \begin{bmatrix} \mathbf{p}_q(x_1, y_1, z_1, n_1) & \mathbf{p}_q(x_2, y_2, z_2, n_2) & \cdots & \mathbf{p}_q(x_M, y_M, z_M, n_M) \end{bmatrix}, \tag{10}$$

$$\mathbf{u} = \begin{pmatrix} u_1 & u_2 & \cdots & u_M \end{pmatrix}, \tag{11}$$

$$\mathbf{q} = \begin{pmatrix} q_1 & q_2 & \cdots & q_M \end{pmatrix}, \tag{12}$$

Substituting Eq. (6) into Eqs. (3) and (4), we obtain:

$$u(x, y, z) = \boldsymbol{\phi}^{uu}(x, y, z)\mathbf{u} + \boldsymbol{\phi}^{uq}(x, y, z)\mathbf{q}, \tag{13}$$

$$q(x, y, z, n) = \boldsymbol{\phi}^{qu}(x, y, z, n)\mathbf{u} + \boldsymbol{\phi}^{qq}(x, y, z, n)\mathbf{q}, \tag{14}$$

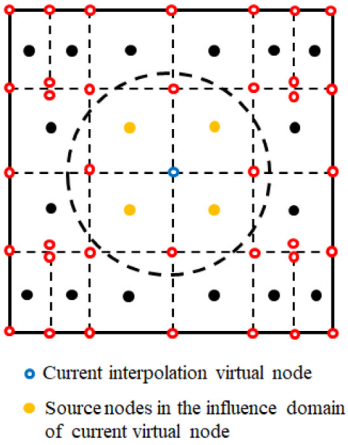


Fig. 6. Schematic diagram of interpolation implementation.

where $\boldsymbol{\phi}^{uu}(x, y, z)$, $\boldsymbol{\phi}^{uq}(x, y, z, n)$, $\boldsymbol{\phi}^{qu}(x, y, z)$, and $\boldsymbol{\phi}^{qq}(x, y, z, n)$ are the shape functions of HMLS interpolation as below:

$$\boldsymbol{\phi}^{uu}(x, y, z) = \mathbf{p}_u^T(x, y, z)\mathbf{C}_t^{-1}\mathbf{P}_u^T, \tag{15}$$

$$\boldsymbol{\phi}^{uq}(x, y, z) = \mathbf{p}_u^T(x, y, z)\mathbf{C}_t^{-1}\mathbf{P}_q^T, \tag{16}$$

$$\boldsymbol{\phi}^{qu}(x, y, z, n) = \mathbf{p}_q^T(x, y, z, n)\mathbf{C}_t^{-1}\mathbf{P}_u^T, \tag{17}$$

$$\boldsymbol{\phi}^{qq}(x, y, z, n) = \mathbf{p}_q^T(x, y, z, n)\mathbf{C}_t^{-1}\mathbf{P}_q^T, \tag{18}$$

Here the HMLS interpolation serves as the second-layer interpolation and its form is defined as follows:

$$u(Q_\beta^v) = \sum_{I=1}^{M} \phi_I^{uu}(x^v, y^v, z^v)\hat{u}(Q_I^s) + \sum_{I=1}^{M} \phi_I^{uq}(x^v, y^v, z^v, n^v)\hat{q}(Q_I^s), \tag{19}$$

$$q(Q_\beta^v) = \sum_{I=1}^{M} \phi_I^{qu}(x^v, y^v, z^v)\hat{u}(Q_I^s) + \sum_{I=1}^{M} \phi_I^{qq}(x^v, y^v, z^v, n^v)\hat{q}(Q_I^s), \tag{20}$$

where M represents the number of the source nodes located in the influence domain (see Fig. 6) of the virtual node Q_β^v, the superscripts s and v denote source and virtual node, respectively. It is worth noting that $\hat{u}(Q_I^s)$ and $\hat{q}(Q_I^s)$ are different from the nodal values of $u(Q_I^s)$ and $q(Q_I^s)$ in general cases.

4. DiBFM with HMLS interpolation for 3-D potential problems

4.1. Boundary integral equation

Consider a potential problem in an arbitrary 3-D domain Ω enclosed with the boundary Γ as shown in Fig. 7:

$$\begin{aligned} &\nabla^2 u = 0, && \forall \mathbf{x} \in \Omega \\ &u = \overline{u}, && \forall \mathbf{x} \in \Gamma_u, \\ &q = \frac{\partial u}{\partial n} = \overline{q}, && \forall \mathbf{x} \in \Gamma_q \end{aligned} \tag{21}$$

where Γ_u and Γ_q are the essential and natural boundaries, respectively. The over bar denotes the prescribed value, and n represents the unit outward normal.

The boundary integral equation for the potential problem is given [26]:

$$c(P)u(P) = \int_\Gamma G(P, Q)q(Q)d\Gamma(Q) - \int_\Gamma F(P, Q)u(Q)d\Gamma(Q), \tag{22}$$

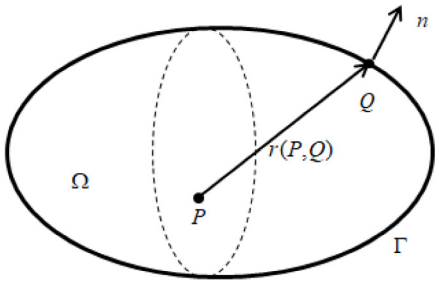


Fig. 7. Objective of arbitrary domain Ω with the boundary Γ.

where P and Q denote the source and field points, respectively, $c(P) = 1/2$ for smooth boundary at P, $G(P, Q)$ represents the response at the field point Q caused by the source P, and $F(P, Q)$ is the normal derivative of $G(P, Q)$, for 3-D potential problems, their forms are given as

$$G(P, Q) = \frac{1}{4\pi r}, \tag{23}$$

$$F(P, Q) = \frac{\partial G(P, Q)}{\partial n(Q)} = -\frac{1}{4\pi r^2} r_{,k}\, n_k(Q), \tag{24}$$

where r is the distance between source point P and field point Q, and $r_{,k} = \partial r/\partial Q_k = (P_k - Q_k)/r$, $k = 1, 2, 3$, n_k denotes the component of normal n.

4.2. Discretization

In the implementation of DiBFM, the dual interpolation elements are applied to discretize the boundary integral Eq. (22). Considering the contribution of virtual nodes, the discretized form of Eq. (22) becomes:

$$\sum_{e=1}^{ne}[\sum_{\alpha=1}^{n\alpha} h^{ss}(P_l)u(Q^s_{e(\alpha)}) + \sum_{\beta=1}^{n\beta} h^{sv}(P_l)u(Q^v_{e(\beta)})] = \sum_{e=1}^{ne}[\sum_{\alpha=1}^{n\alpha} g^{ss}(P_l)q(Q^s_{e(\alpha)}) + \sum_{\beta=1}^{n\beta} g^{sv}(P_l)q(Q^v_{e(\beta)})], \tag{25}$$

in which

$$h^{ss}(P_l) = \int_{\Gamma_e} \frac{\partial G(P, Q)}{\partial n(Q)} N^s_{e(\alpha)}(Q) d\Gamma(Q) + \frac{1}{2}\delta^l_{e(\alpha)}, \tag{26}$$

$$h^{sv}(P_l) = \int_{\Gamma_e} \frac{\partial G(P, Q)}{\partial n(Q)} N^v_{e(\beta)}(Q) d\Gamma(Q), \tag{27}$$

$$g^{ss}(P_l) = \int_{\Gamma_e} G(P, Q) N^s_{e(\alpha)}(Q) d\Gamma(Q), \tag{28}$$

$$g^{sv}(P_l) = \int_{\Gamma_e} G(P, Q) N^v_{e(\beta)}(Q) d\Gamma(Q), \tag{29}$$

and

$$\delta^l_{e(\alpha)} = \begin{cases} 1, & \text{if } P_l \text{ is located at } \alpha\text{th source node in } e\text{th element} \\ 0 & \end{cases}, \tag{30}$$

where P_l denote lth source point (l= 1, 2, …, Ns). Ns is the number of source points, ne is the number of elements and $n\alpha$, $n\beta$ indicate the number of source and virtual nodes in an element, respectively. $N^s_{e(\alpha)}$ and $u(Q^s_{e(\alpha)})$ represent the shape function and potential of the αth source point in eth element.

In Eqs. (26)–(29), singular and nearly singular integrals arise when the source point is on or close to the integration element. Since only the source points are arranged as the collocation points in dual interpolation element, the integration scheme is the same as that for traditional elements, for which various methods have been put forward to compute these integrals [27,28]. In this article, a spherical element subdivision method (SSM) [29] is employed to

obtain integral patches in suitable size and shape, and the method using approximate expansion of local coordinate is applied to handle the hypersingular integral [30].

4.3. Assembly and solution

Eq. (25) is expressed in matrix form as follows:

$$\mathbf{Hu} = \mathbf{Gq}, \tag{31}$$

where $\mathbf{u}$ and $\mathbf{q}$ denote the vectors of physical variables. $\mathbf{H}$ and $\mathbf{G}$ are the coefficient matrices obtained by boundary integrals.

Distinguishing the sources points from the virtual points in Eq. (31), we obtain:

$$\begin{bmatrix}\mathbf{H}^{ss} & \mathbf{H}^{sv}\end{bmatrix}\begin{Bmatrix}\mathbf{u}^{s}\\ \mathbf{u}^{v}\end{Bmatrix} = \begin{bmatrix}\mathbf{G}^{ss} & \mathbf{G}^{sv}\end{bmatrix}\begin{Bmatrix}\mathbf{q}^{s}\\ \mathbf{q}^{v}\end{Bmatrix}, \tag{32}$$

where the superscripts s and v denote the source and virtual points, respectively. Furthermore, the vectors $\mathbf{u}^{v}$ and $\mathbf{q}^{v}$ can be rewritten as:

$$\mathbf{u}^{\mathbf{v}} = \overline{\mathbf{u}}^{\mathbf{v}} + \widetilde{\mathbf{u}}^{\mathbf{v}}, \tag{33}$$

$$\mathbf{q}^{\mathbf{v}} = \overline{\mathbf{q}}^{\mathbf{v}} + \widetilde{\mathbf{q}}^{\mathbf{v}}, \tag{34}$$

where the superscripts '$-$' and '$\sim$' denote the boundaries with and without specified values, respectively. The unknown potentials and fluxes can be approximated by HMLS approximation discussed above. Therefore, $\widetilde{\mathbf{u}}^{\mathbf{v}}$ and $\widetilde{\mathbf{q}}^{\mathbf{v}}$ are given as shown below:

$$\widetilde{\mathbf{u}}^{\mathbf{v}} = \phi_{uu}^{vs}\mathbf{u}^{s} + \phi_{uq}^{vs}\mathbf{q}^{s}, \tag{35}$$

$$\widetilde{\mathbf{q}}^{\mathbf{v}} = \phi_{qu}^{vs}\mathbf{u}^{s} + \phi_{qq}^{vs}\mathbf{q}^{s}, \tag{36}$$

where ϕ_{uu}^{vs}, ϕ_{uq}^{vs}, ϕ_{qu}^{vs}, and ϕ_{qq}^{vs} are the shape function matrices of HMLS approximation defined above (Section 3.2).

Substituting Eqs. (33)–(36) into Eq. (32), we obtain:

$$\mathbf{H}^{ss}\mathbf{u}^{s} + \mathbf{H}^{sv}\boldsymbol{\Phi}_{uu}^{vs}\mathbf{u}^{s} + \mathbf{H}^{sv}\boldsymbol{\Phi}_{uq}^{vs}\mathbf{q}^{s} + \mathbf{H}^{sv}\overline{\mathbf{u}}^{v} = \mathbf{G}^{ss}\mathbf{q}^{s} + \mathbf{G}^{sv}\boldsymbol{\Phi}_{qu}^{vs}\mathbf{u}^{s} + \mathbf{G}^{sv}\boldsymbol{\Phi}_{qq}^{vs}\mathbf{q}^{s} + \mathbf{G}^{sv}\overline{\mathbf{q}}^{v}, \tag{37}$$

and with simplification, we have:

$$\mathbf{H}^{s}\mathbf{u}^{s} = \mathbf{G}^{s}\mathbf{q}^{s} + \mathbf{b}^{v}, \tag{38}$$

where

$$\mathbf{H}^{s} = \mathbf{H}^{ss} + \mathbf{H}^{sv}\boldsymbol{\Phi}_{uu}^{vs} - \mathbf{G}^{sv}\boldsymbol{\Phi}_{qu}^{vs}, \tag{39}$$

$$\mathbf{G}^{s} = \mathbf{G}^{ss} + \mathbf{G}^{sv}\boldsymbol{\Phi}_{qq}^{vs} - \mathbf{H}^{sv}\boldsymbol{\Phi}_{uq}^{vs}, \tag{40}$$

$$\mathbf{b}^{v} = -\mathbf{H}^{sv}\overline{\mathbf{u}}^{v} + \mathbf{G}^{sv}\overline{\mathbf{q}}^{v}. \tag{41}$$

Considering the boundary conditions, Eq. (38) gives:

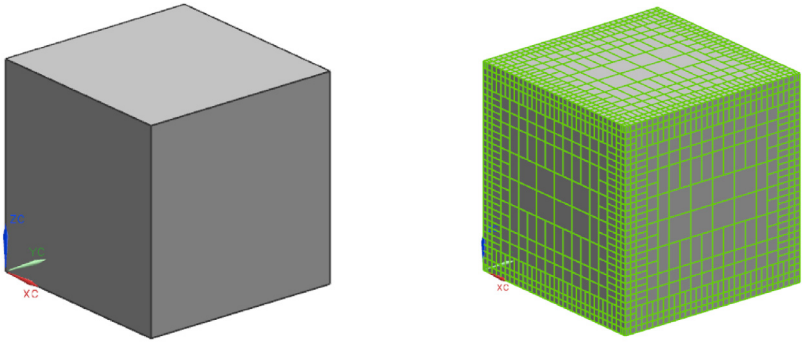
$$\begin{Bmatrix}\overline{\mathbf{H}}^{\mathbf{s}} & \widetilde{\mathbf{H}}^{\mathbf{s}}\end{Bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}^{\mathbf{s}}\\ \widetilde{\mathbf{u}}^{\mathbf{s}}\end{Bmatrix} = \begin{Bmatrix}\overline{\mathbf{G}}^{\mathbf{s}} & \widetilde{\mathbf{G}}^{\mathbf{s}}\end{Bmatrix}\begin{Bmatrix}\overline{\mathbf{q}}^{\mathbf{s}}\\ \widetilde{\mathbf{q}}^{\mathbf{s}}\end{Bmatrix} + \mathbf{b}^{\mathbf{v}}, \tag{42}$$

Finally a linear algebraic equation is given as:

$$\mathbf{Ax} = \mathbf{b}, \tag{43}$$

where

$$\mathbf{A} = \begin{Bmatrix}\widetilde{\mathbf{H}}^{\mathbf{s}} & \widetilde{\mathbf{G}}^{\mathbf{s}}\end{Bmatrix}, \mathbf{x} = \begin{Bmatrix}\widetilde{\mathbf{u}}^{\mathbf{s}}\\ \widetilde{\mathbf{q}}^{\mathbf{s}}\end{Bmatrix}, \mathbf{b} = \overline{\mathbf{G}}^{s}\overline{\mathbf{q}}^{s} + \mathbf{b}^{v} - \overline{\mathbf{H}}^{s}\overline{\mathbf{u}}^{s} \tag{44}$$



(a) (b)

Fig. 8. (a) Geometry of a cube and (b) its mesh with binary tree grids.

in which **A** is the coefficient matrix of linear system of equation, **x** is the unknown vector only at source points, and **b** is the vector obtained by boundary condition.

Eq. (31) is of same scale as the conventional BEM, however, the DiBFM can achieve higher accuracy by increasing the interpolation precision.

5. Numerical examples

In this section, we present four numerical examples to demonstrate the superiority of the proposed method. Firstly, the example of cube with analytical solution is used to test the accuracy and efficiency. The next two examples are presented to show its advantages in solving thin-wall structures. The last example demonstrates its capacity to deal with complex model. Analytical solutions and solutions given by the finite element software are compared in all examples. In addition, the relative errors are considered to evaluate the accuracy and convergence of DiBFM, which are defined as:

$$error = \frac{1}{\left|w^{(e)}\right|_{\max}} \sqrt{\frac{1}{M}\sum_{i=1}^{M}[w_i^{(e)} - w_i^{(n)}]^2}, \tag{45}$$

where $\left|w^{(e)}\right|_{\max}$ denote either maximum absolute value of potential or its normal derivative over M sample points, $u_i^{(e)}$ and $u_i^{(n)}$ represent the reference solutions.

5.1. Cube with mixed boundary condition

The first example is a cube of the unit length as shown in Fig. 8. The Dirichlet boundary condition is applied on the faces of $z = \pm 1$ and the Neumann boundary condition is applied on the remaining faces. The potential and normal flux boundary conditions are given as:

$$u(\mathbf{x}) = x^3 + y^3 + z^3 - 3x^2y - 3y^2z - 3z^2x \tag{46}$$

$$q(\mathbf{x}) = -k\frac{\partial u(\mathbf{x})}{\partial n(\mathbf{x})} = -k(q_x n_x + q_y n_y + q_z n_z) \tag{47}$$

where

$$q_x = 3x^2 - 6xy - 3z^2, q_y = 3y^2 - 6yz - 3x^2, q_z = 3z^2 - 6xz - 3y^2 \tag{48}$$

where **x** represents the field point in 3-D Cartesian coordinate system. $n(\mathbf{x})$ or (n_x, n_y, n_z) denote the outward normal, $k = 1$ W/(m K) is the heat conductivity.

The results given by the tradition BEM are presented. Symbols *Err_u* and *Err_q* represent the relative errors of potential and normal flux, *Ns* and *Time* denote the number of source nodes and the total CPU time, respectively.

Table 1
Comparison of relative errors between DiBFM and Traditional BEM.

DiBFM			Tradition BEM		
Ns	*Err_u*	*Err_q*	*Ns*	*Err_u*	*Err_q*
136	9.03E−4	2.43E−2	136	7.18E−3	2.78E−2
400	2.80E−4	1.14E−2	406	5.19E−3	2.42E−2
880	9.85E−5	5.27E−3	934	2.63E−3	1.62E−2
2416	3.12E−5	1.75E−3	2566	1.15E−4	9.79E−3

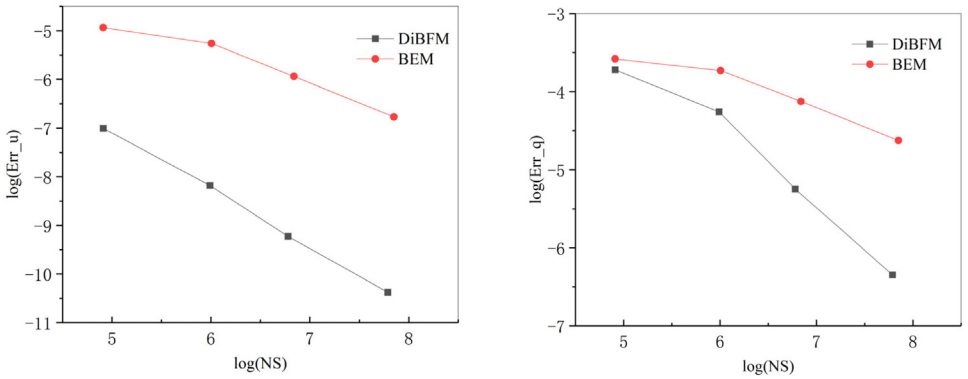


Fig. 9. Comparison of accuracy between DiBFM and Tradition BEM.

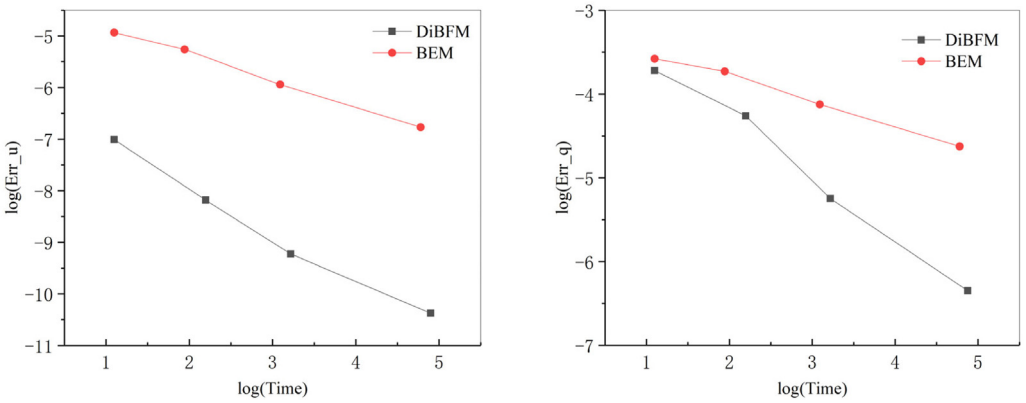


Fig. 10. Comparison of efficiency between DiBFM and Traditional BEM.

The relative errors *Err_u* and *Err_q* are presented in Table 1. The comparison of accuracy and efficiency between DiBFM and Tradition BEM is depicted in Figs. 9 and 10.

From the results above, It is seen that the accuracy and efficiency of DiBFM are both better than that of the tradition BEM under the condition that the same number of nodes are used, and the convergence speed of DiBFM is much faster with the increasing number of nodes.

5.2. Goblet with small feature structure

The second example is from the steady heat conduction problem of a goblet with small feature structure shown in Fig. 11(a). The goblet is filled with 50 °C water and its outer surfaces are exposed to air. So the Dirichlet boundary conditions are applied to the inner surfaces of the goblet and the remaining surfaces are specified with the Robin boundary conditions. The environment temperature, the heat conductivity and convection are chosen to

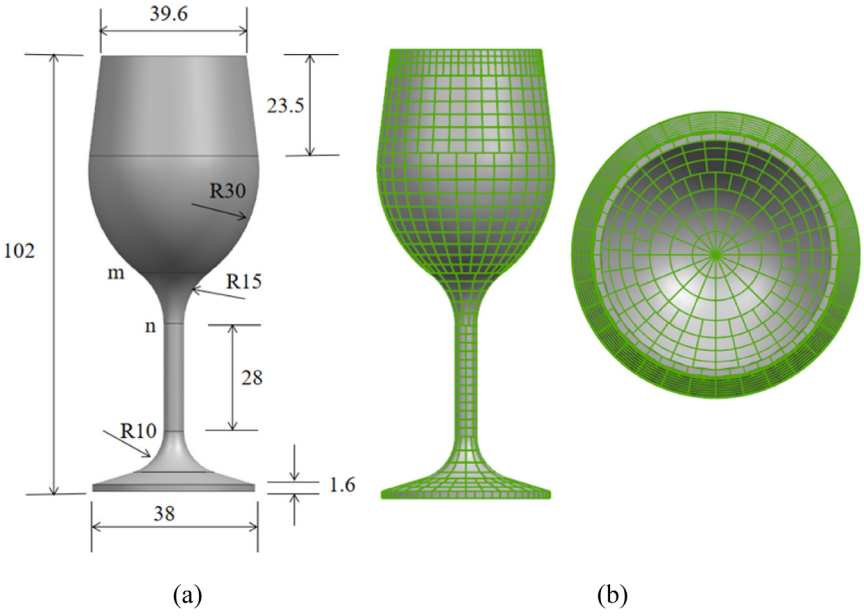


Fig. 11. (a) Goblet geometric model and (b) its mesh with binary tree grids.

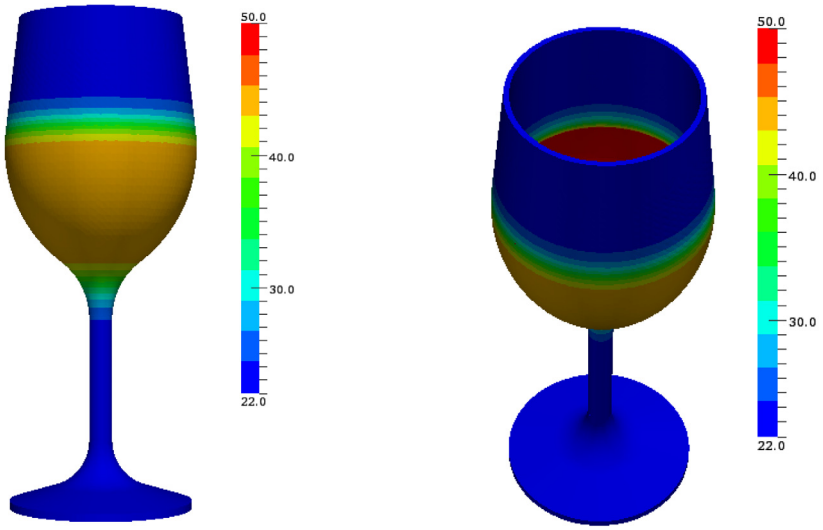


Fig. 12. Contours of temperature: DiBFM with 1880 source nodes.

22 °C, 7.6 W/(m K) and 0.8. The domain is discretized with binary tree grids shown in Fig. 11(b) with constant elements in DiBFM.

The temperature distribution of the goblet obtained by DiBFM with 1880 nodes is shown in Fig. 12, and the results given by FEM with 222335 nodes are presented in Fig. 13 for comparison.

Results along the curve m-n by the two methods are shown in Fig. 14, Y denotes the vertical distance from the observing points.

It can be seen that the contours given by our method using fewer nodes are consistent with FEM using much more nodes. And the result obtained by DiBFM on the curve m-n is almost identical to FEM. This example also validates the applicability of DiBFM to complicated structures with small geometric features.

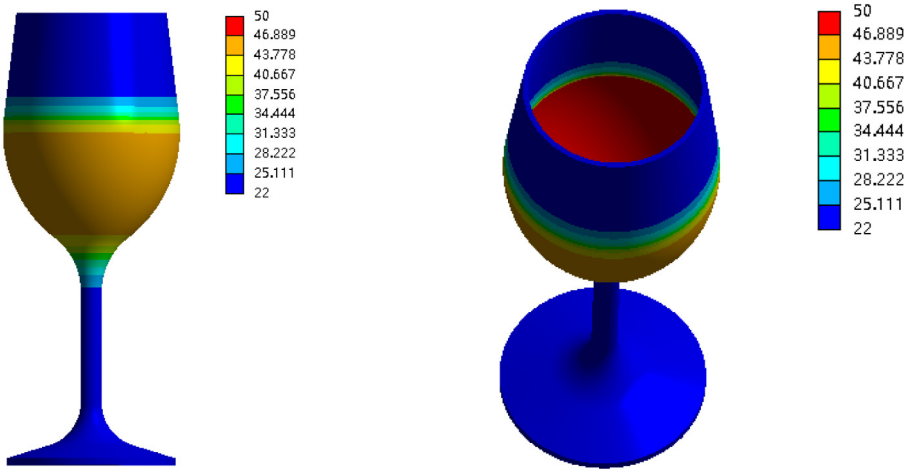


Fig. 13. Contours of temperature: FEM with 222335 nodes.

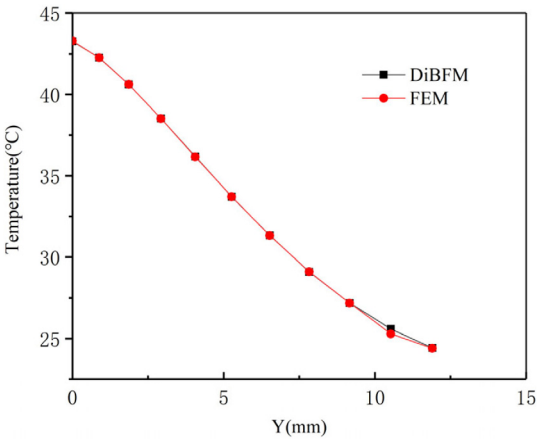


Fig. 14. Comparison of temperature result along the curve **m-n** of DiBFM and FEM.

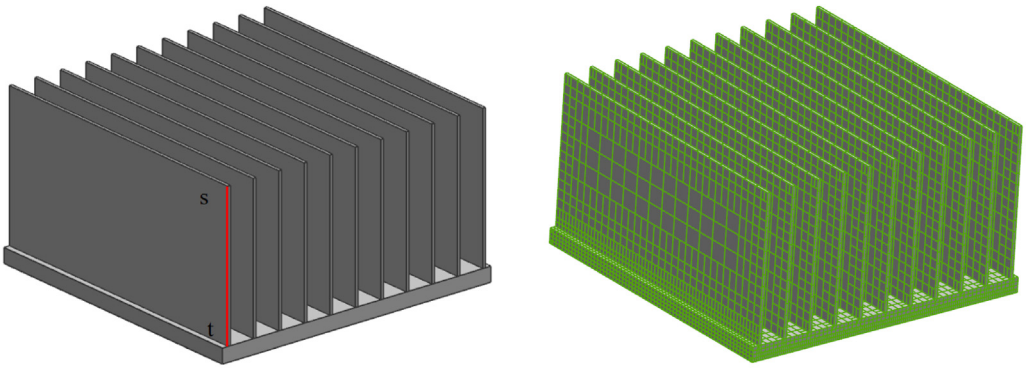
5.3. Sink with thin-walls

Next, we consider a mixed boundary condition problem of a sink with thin-walls shown in Fig. 15(a). The Dirichlet boundary condition is specified on the sub-surface and, Neumann and Robin boundary conditions on the base and the remaining faces of the sink, respectively. The heat flux, the environment temperature, the heat conductivity and convection are selected as 10 W, 22 °C, 237 W/(m K) and 0.8.

The contours of temperature in the goblet obtained by DiBFM with 14393 nodes are given in Fig. 16 and the contours given by FEM using 524133 nodes are shown in Fig. 17 for comparison. The temperature variations along the straight line **s-t** are plotted in Fig. 18, where Z represents the vertical distance from the point **s**. Excellent agreement can be obtained. This example demonstrates that DiBFM with HMLS interpolation is applicable to thin-wall structures.

5.4. Sleeve

To demonstrate the ability to deal with the arbitrary complex geometry, we investigate the steady heat conduction in a sleeve as shown in Figs. 19–20 with mixed boundary conditions. The hole surfaces of the sleeve are specified with the Dirichlet boundary conditions. Temperatures on the surfaces of the middle hole are 100 °C and 60 °C on



(a) (b)

Fig. 15. (a) Geometric model of a sink and (b) its mesh with binary tree grids.

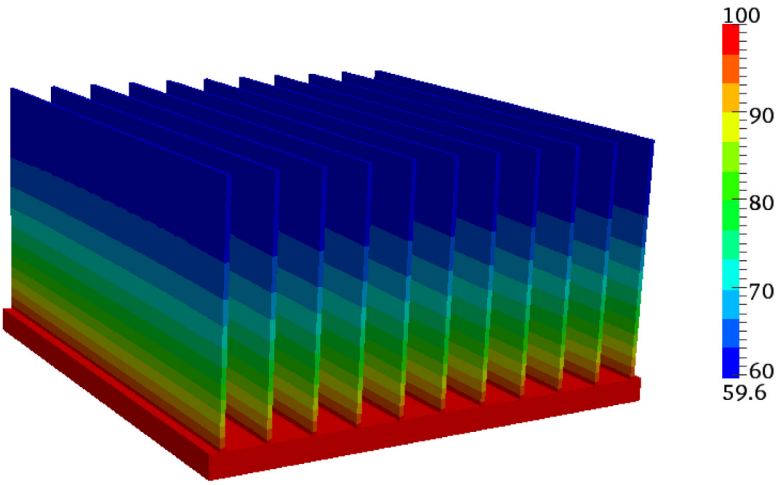


Fig. 16. Contours of temperature: DiBFM with 14393 source nodes.

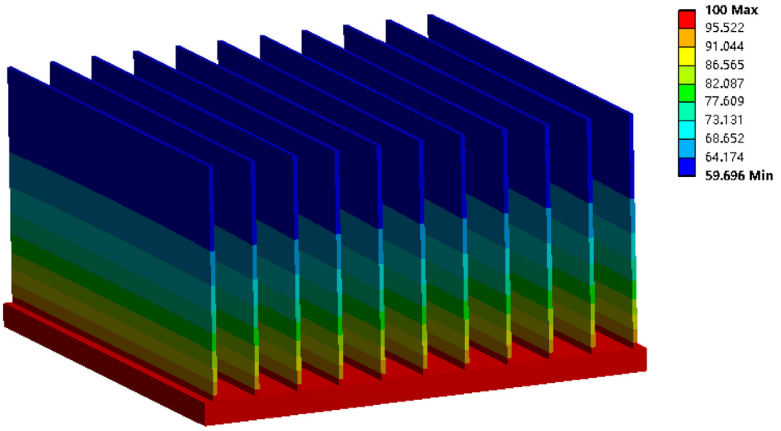


Fig. 17. Contours of temperature: FEM with 524133 nodes.

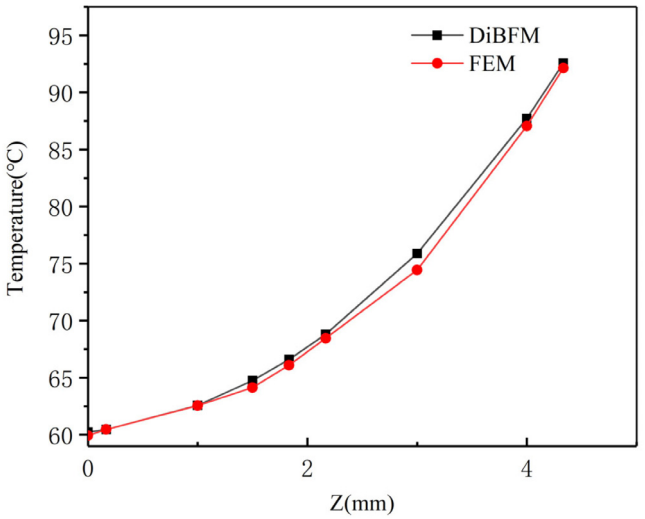


Fig. 18. Temperature variations along straight line **s-t** with DiBFM and FEM.

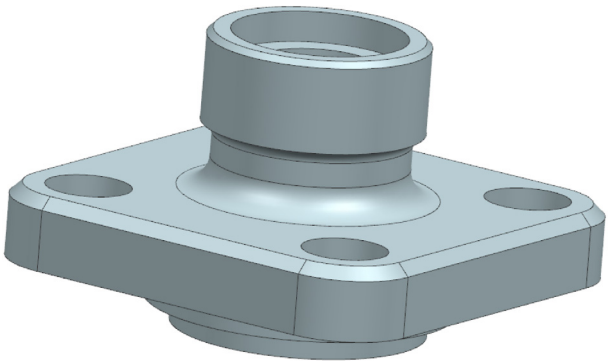


Fig. 19. Geometry of sleeve.

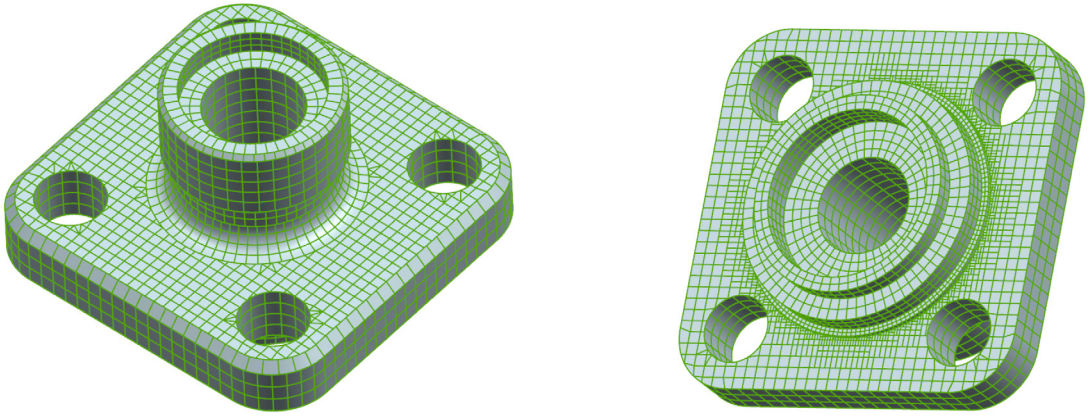


Fig. 20. Mesh with binary tree grids for sleeve.

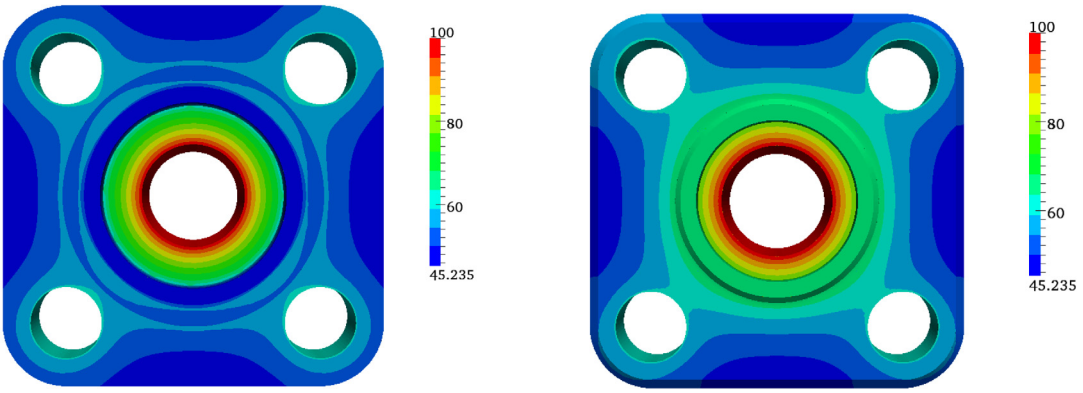


Fig. 21. Contours of temperature: DiBFM with 5324 source nodes.

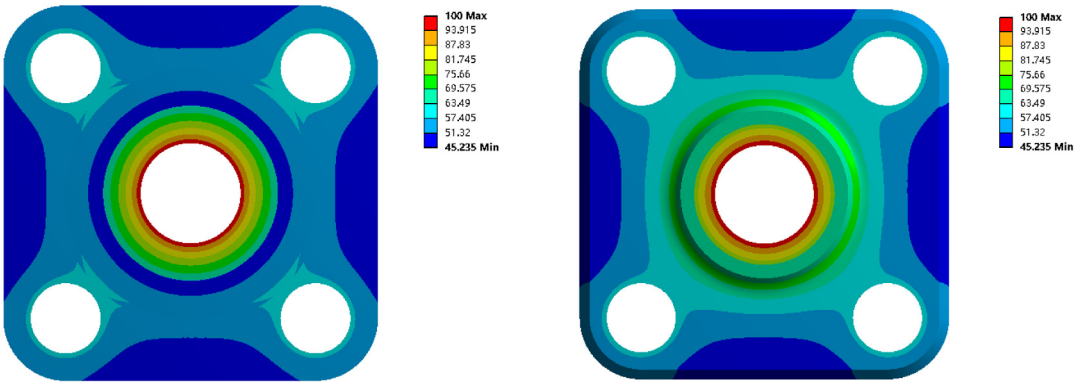


Fig. 22. Contours of temperature: FEM with 964528 nodes.

the other four small hole surfaces. Robin boundary conditions are applied to the remaining surfaces of the sleeve. The environment temperature, heat conductivity and convection are chosen 22 °C, 40 W/(m K) and 0.8, respectively and constant elements are used.

Fig. 21 shows the temperature variations obtained by DiBFM with 5324 nodes and the results by FEM with 964528 element nodes are shown in Fig. 22 for comparison. By observing these two solutions, we can see that the results by those two methods are almost the same which shows the accuracy of DiBFM for solving general engineering problems.

6. Discussion and conclusions

The DiBFM coupled with the binary tree grids scheme has been extended to 3-D potential problems. In the implementation of the DiBFM, CAE analysis is performed directly on CAD models, the geometric errors between CAD and CAE model are avoided. By adding virtual nodes on traditional discontinuous element, the dual interpolation elements are introduced. The first-layer interpolation is applied to approximate the physical variable by source and virtual nodes, and the meshless interpolation is constructed to interpolate the values of the virtual nodes with the source in the second-layer interpolation. Compared with the traditional discontinuous element, the dual interpolation element has higher interpolation accuracy and can be applied to continuous and discontinuous field. And it also takes slightly more time due to the integration and assembly for virtual nodes.

Since the BTSM can generate the continuous and discontinuous grids automatically in general, automatic grids generation of complex models becomes much easier. And the ability to use discontinuous grids for efficient calculations is a major advantage of our method. However, we have to admit that using continuous grids can achieve higher accuracy than using discontinuous grids when only the grid continuity is considered.

Numerical examples for potential problems demonstrate that the DiBFM with discontinuous grids has higher accuracy and efficiency than the conventional BEM. Compared with the FEM, this method can achieve same accuracy degree much less nodes and has faster convergence rate. More complex 3-D engineering problems such as acoustic simulations can be observed by DiBFM in the future work.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: JianMing Zhang reports financial support was provided by National Natural Science Foundation of China. Jianming Zhang reports a relationship with Hunan University that includes: employment.

Acknowledgments

This work was supported by National Natural Science Foundation of China under grant numbers 11772125 and 11972010.

References

[1] C.A. Brebbia, The Boundary Element Method for Engineers [M], Pentech Press, 1984.
[2] L. Shen, Y. Liu, A an adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton–Miller formulation, Comput. Mech. 60 (2007) 461–472.
[3] A. Buffa, G. Sangalli, et al., Isogeometric analysis in electromagnetics: B-splines approximation, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1143–1152.
[4] J.M. Zhang, W.C. Lin, et al., A dual interpolation boundary face method for exterior acoustic problems based on the Burton–Miller formulation, Eng. Anal. Bound. Elem. 113 (2020) 219–231.
[5] C.Y. Dong, X. Yang, E. Pan, Analysis of cracked transversely isotropic and inhomogeneous solids by a special BIE formulation, Eng. Anal. Bound. Elem. 35 (2011) 200–206.
[6] 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.
[7] J.M. Zhang, L. Yang, et al., A dual interpolation Galerkin boundary face method for potential problems, Eng. Anal. Bound. Elem. 117 (2020) 157–166.
[8] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4135–4195.
[9] A. Buffa, G. Sangalli, R. vazquez, Isogeometric analysis in electromagnetics:B-splines approximation, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1143–1152.
[10] F. Auricchio, B. Daveiga, A. Buffa, A fully locking-free isogeometric approach for plane linear elasticity problems, Comput. Methods Appl. Mech. Engrg. 197 (2007) 160–172.
[11] R.N. Simpson, S.P.A. Bordas, et al., A two-dimensional isogeometric boundary element method for elastostatic analysis, Comput. Methods Appl. Mech. Eng. 209-212 (2012) 87–100.
[12] M.J. Peake, J. Trevelyan, G. Coater, Extended isogeometric boundary element method(XIBEM) for two-dimensional Helmholtz problems, Comput. Methods Appl. Mech. Engrg. 259 (2013) 93–102.
[13] Y. Bazilevs, V.M. Calo, J.A. Cottrell, et al., Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Engrg. 199 (2010) 229–263.
[14] J.M. Zhang, X.Y. Qin, X. Han, Y. Li, A boundary face method for potential problems in three dimensions, Internat. J. Numer. Methods Engrg. 80 (2009) 320–377.
[15] J.M. Zhang, W.C. Lin, et al., A dual interpolation boundary face method for elasticity problems, Eur. J. Mech. A Solids 73 (2019) 500–511.
[16] W.F. Florez, H. Powe, Comparison between continuous and discontinuous bound- ary elements in the multidomain dual reciprocity method for the solution of the two-dimensional Navier–Stokes equations, Eng. Anal. Bound. Elem. 25 (2001) 57–69.
[17] J.M. Zhang, W.C. Lin, Y.Q. Dong, C.M. Ju, A double-layer interpolation method for implementation of BEM analysis of problems in potential theory, Appl. Math. Model. 51 (2017) 250–269.
[18] J.M. Zhang, C.M. Ju, et al., A binary-tree subdivision method for evaluation of singular integrals in 3-D BEM, Eng. Anal. Bound. Elem. 103 (2019) 80–93.
[19] J.R. Tristano, S.J. Owen. S.A. Canann, Advancing front surface mesh generation in parametric space using a riemannian surface definition, in: Proceedings of 7th International Meshing Roundtable, 1998, pp. 429-445.
[20] T.P. Fang, L.A. Piegl, Delaunay triangulation using a uniform grid, IEEE Comput. Graph. Appl. 13 (1993) 36–47.
[21] K.H. Baldwin, H.L. Schreyer, Automatic generation of quadrilateral elements by a conformal mapping, Eng. Comput. 2 (1985) 187–194.
[22] G. Li, N.R. Aluru, Boundary cloud method: a combined scattered point/boundary integral approach for boundary-only analysis, Comput. Methods Appl. Mech. Engrg. 191 (2002) 2337–2370.
[23] G. Li, N.R. Aluru, A boundary cloud method with a cloud-by-cloud polynomial basis, Eng. Anal. Bound. Elem. 27 (2003) 57–71.
[24] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, Math. Comp. 37 (155) (1981) 141–158.

[25] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, IMA J. Numer. Anal. 32 (2012) 983–1000.
[26] J.M. Zhang, B.T. Chi, et al., A dual interpolation boundary face method for three-dimensional potential problems. International, J. Heat Mass Transfer 140 (2019) 862–876.
[27] Hang. Ma, Norio. Kamiya, Distance transformation for the numerical evaluation of near singular boundary integrals with various kernels in boundary element method, Eng. Anal. Bound. Elem. 26 (2002) 329–339.
[28] X.W. Gao, An effective method for numerical evaluation of general 2D and 3D high order singular boundary integrals, Comput. Methods Appl. Mech. Engrg. 199 (2010) 2856–2864.
[29] J.M. Zhang, P. Wang, C.J. Lu, Y.Q. Dong, A spherical element subdivision method for the numerical evaluation of nearly singular integrals in 3D BEM, Eng. Comput. 34 (6) (2017) 2074–2087.
[30] 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 (3) (1992) 604–614.