Engineering Analysis with Boundary Elements 162 (2024) 157–171

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

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

Comparative study on domain decomposition methods for solving multi-domain potential problems by DiBFM

Rongxiong Xiao [a], Jianming Zhang [a,*], Yang Yang [b], Chong Zhang [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] Shenzhen MSU-BIT University, Shenzhen, Guangdong, China

ARTICLE INFO

Keywords:
Non-overlapping domain decomposition methods
DiBFM
Multi-domain problems

ABSTRACT

Domain decomposition method (DDM) is an efficient tool for solving multi-domain problems. Many kinds of DDM have been proposed in literature, such as the D-N alternating method and optimized Schwarz method, etc. The main difference between these methods lies in the transmission conditions on the interfaces between subdomains. Most DDMs have at least one relaxation parameter which must be properly chosen to ensure convergence. To avoid prior determination of a constant relaxation parameter, dynamic relaxation parameters are devised based on error analysis. Another, difficulty of DDM is that special preprocessor has to be used to handle cases where interior cross-points exist among the domains. In dual interpolation boundary face method (DiBFM), boundary integral equation (BIE) is collocated on nodes inside elements, therefore naturally avoids the interior cross-points problem. In this paper, we have implemented some main kinds of DDM in the frame of DiBFM and conducted comparative studies accounting for different interface conditions and relaxation parameters, using examples that involve different topological connections between subdomains.

1. Introduction

In recent decades, multi-domain problems have been intensively studied and applied in various fields, such as potential problems [1,2], elasticity problems [3,4], nonhomogeneous material problems [5], etc. There are two main approaches to solve multi-domain problems. One is to directly assemble the equations of all the subdomains to form an overall system of linear equations [1,5,6]. Although the coefficient matrix exhibits some sparsity which can be used to improve efficiency, the required computation time and memory are still large, and the filling problem in direct solvers for sparse matrices is also troublesome. The other is the so-called domain decomposition method (DDM) which firstly proposed by Schwarz [7]. In DDM, the governing equations are separately set up for each subdomain, the relationships between the subdomains are established by interface conditions, and each subdomain is solved independently. Therefore, DDM can significantly reduce CPU time and memory usage, and is easy to parallelize. Due to its own advantages, DDM has been developed rapidly in recent years. Many kinds of DDM have been proposed, such as the well-known D-N alternating method [8] and P.L. Lions method [9,10], the main difference between these methods lies in the interface conditions. In the implementation of D-N alternating method, the Dirichlet and Neumann boundary conditions are specified on the interfaces, respectively, while the Robin boundary condition is used in P.L. Lions method. In addition, the DDMs using Dirichlet-Dirichlet, Neumann-Neumann interface conditions can also be found in [11–13].

By combining with different numerical methods, such as the finite element method (FEM) and the boundary element method (BEM), DDMs are widely used in solving various multi-domain problems [14–18]. Contrary to direct methods, two aspects should be specially noticed in DDMs. First, DDMs are iterative methods and the relaxation parameters should be predetermined in the iterative equations. A reasonable relaxation parameter can ensure or accelerate convergence. However, the choice of relaxation parameter is problem dependent. To avoid prior determination of relaxation parameter, dynamic relaxation parameters can be devised based on error analysis in each iterative step. The second aspect is that DDMs cannot guarantee convergence when there are interior cross-points among subdomains which inevitably exist in real engineering structures. Usually, a special preprocessor has to be constructed to deal with the interior cross-points in the implementation of DDMs. Fortunately, in dual interpolation boundary face method (DiBFM) [19], boundary integral equation (BIE) is collocated on nodes

* Corresponding author.
E-mail address: zhangjm@hnu.edu.cn (J. Zhang).

https://doi.org/10.1016/j.enganabound.2024.01.011
Received 23 November 2023; Received in revised form 14 January 2024; Accepted 14 January 2024
Available online 13 February 2024

Table 1
The variable description information table.

Variables	Description
ξ, η	Parameter coordinates
$N_i^{s(v)}$	Element shape function of the *i th* source node/virtual node
Superscripts '*s*', '*v*'	Source node /virtual node
$\mathbf{UU}^{vs}$,$\mathbf{UQ}^{vs}$,$\mathbf{QU}^{vs}$, $\mathbf{QQ}^{vs}$	Shape functions in meshless method
$\mathbf{n}$	Unit outward normal
Ω_i	The *i th* subdomain
Γ_u,Γ_q,Γ_I	Dirichlet, Neumann and interface boundaries
$\mathbf{H}_{dn}^{sv}$, $\mathbf{G}_{dn}^{sv}$	Coefficient matrix blocks obtained by integrating the source nodes at the Dirichlet boundaries with the virtual nodes at the Neumann boundaries.
$\widetilde{\alpha}_r^{s(v)}$,$\widetilde{\beta}_r^{s(v)}$	Vectors with constant Robin coefficients
Subscripts '*M*', '*S*'	Master/ slave interface
nItr	The *n-th* iteration/Number of iterations
α, β, γ, ρ	Relaxation parameters in DDMs
λ	Coupling coefficient in parallel Robin- Robin DDM
ε	Tolerance
Err_u, Err_q	The relative errors of temperature and flux

inside elements. The interior cross-points problem can be naturally avoided. Besides, the trial function need not to be continuous in DiBFM, discontinuous meshes [20] can be employed to discrete the boundaries, which reduces the difficulty of automatic mesh generation for complex models. In this paper, five main DDMs including different interface conditions and relaxation parameters are implemented into the frame of DiBFM and comparative studies on these methods are conducted by using a number of examples with different topological connections between subdomains.

This paper is arranged as follows. Section 2 derives the DiBFM for multi-domain problems and its discrete scheme. Section 3 introduces five non-overlapping DDMs with DiBFM. Numerical examples are given in Section 4. The conclusions are researched in Section 5. The description information table for main variables in this paper is given below (Table 1).

2. DiBFM for multi-domain potential problems

2.1. DiBFM

The boundary face method (BFM) [21] inherits the advantages of the BEM and has its own characteristics. Different from isogemetric analysis method [22], in the implementation of the BFM, CAE analysis is directly performed on CAD model, thus CAE and CAD are naturally integrated. As shown in Fig. 1, the BFM model is completely consistent with the original CAD model, while the BEM model has geometric errors.

Introducing the dual interpolation element, there are two kinds nodes as shown in Fig. 2. One is the source node located inside the element and the other is virtual node arranged at the vertices and edges of the element. Since only the source nodes are used for the collocation points of boundary integral equation, the situation of interior cross-points can be avoided.

There are two layers of interpolation in dual interpolation elements, the first-layer interpolation is used to approximate physical quantities and its forms are given below:

$$u(\xi,\eta)=\sum_{i=1}^{n_i}N_i^s(\xi,\eta)u\left(Q_i^s\right)+\sum_{j=1}^{n_j}N_j^v(\xi,\eta)u\left(Q_j^v\right) \tag{1}$$

$$q(\xi,\eta)=\sum_{i=1}^{n_i}N_i^s(\xi,\eta)q\left(Q_i^s\right)+\sum_{j=1}^{n_j}N_j^v(\xi,\eta)q\left(Q_j^v\right) \tag{2}$$

The second-layer interpolation uses the meshless method to condense the degree of freedom (DOF) of virtual nodes, and its forms are defined below:

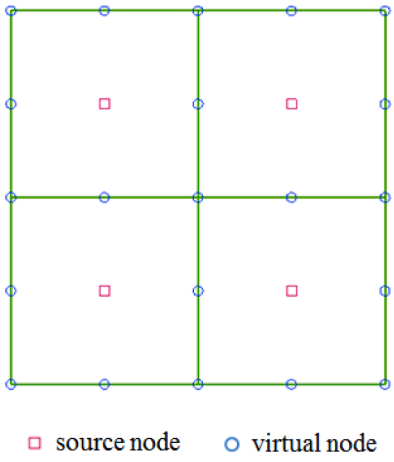

Fig. 2. Dual interpolation constant elements.

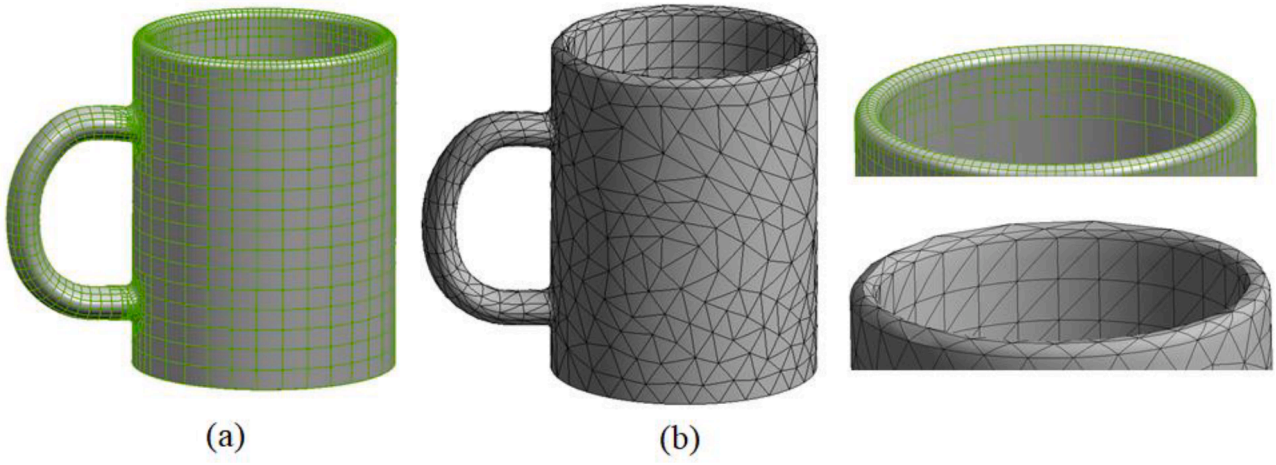

Fig. 1. (a) BFM model (b) BEM model.

$$u\left(Q_\beta^v\right)=\sum_{I=1}^{M}\mathbf{UU}^{vs}(x^v,y^v,z^v)\widehat{u}\left(Q_I^s\right)+\sum_{I=1}^{M}\mathbf{UQ}^{vs}(x^v,y^v,z^v,n^v)\widehat{q}\left(Q_I^s\right) \tag{3}$$

$$q\left(Q_\beta^v\right)=\sum_{I=1}^{M}\mathbf{QU}^{vs}(x^v,y^v,z^v)\widehat{u}\left(Q_I^s\right)+\sum_{I=1}^{M}\mathbf{QQ}^{vs}(x^v,y^v,z^v,n^v)\widehat{q}\left(Q_I^s\right) \tag{4}$$

where (ξ, η) and (*x, y, z*) denote the surface parameter and Cartesian coordinates respectively. The superscripts '*s*' and '*v*' represent the source and virtual nodes respectively. ***N*** are the shape functions of element and $\mathbf{UU}^{vs}$,$\mathbf{UQ}^{vs}$,$\mathbf{QU}^{vs}$,$\mathbf{QQ}^{vs}$ denote the shape functions in meshless method respectively. **n** is the unit outward normal. n_i, n_j are the number of source and virtual nodes in a dual interpolation element respectively. *M* is the number of source nodes located in the influence domain of the virtual node Q_β^v as shown in Fig. 3.

2.2. Boundary integral equation for multi-domain potential problem

Consider a multi-domain potential problem as shown in Fig. 4, we have:

$$\begin{aligned}&\nabla^2 u=0, && \forall \mathbf{x}\in\Omega_i (i=1,2,3)\\ &u=\overline{u}, && \forall \mathbf{x}\in\Gamma_u\\ &q=\frac{\partial u}{\partial n}=\overline{q}, && \forall \mathbf{x}\in\Gamma_q\end{aligned} \tag{5}$$

where Ω_i denotes the *i* th subdomain, *u*and q are the potential and normal flux, respectively. Γ_u, $\Gamma_q$and $\Gamma_I$represent the Dirichlet, Neumann and interface boundaries, respectively.

For each subdomain, the boundary integral equation is configured separately as follows:

$$c(P)u(P)=\int_\Gamma G(P,Q)q(Q)d\Gamma(Q)-\int_\Gamma F(P,Q)u(Q)d\Gamma(Q) \tag{6}$$

where *P* and Q represent the source and field point respectively, *G*(*P, Q*) and *F*(*P, Q*) are the kernel functions. *c*(*P*) is a coefficient and equals to 1/2 when the boundary around *P* is smooth.

2.3. Discretization and assembly

The Discretization of Eq. (6) using dual interpolation elements can be written in matrix form as shown below:

$$\begin{bmatrix}\mathbf{H}_{dd}^{ss}&\mathbf{H}_{dr}^{ss}&\mathbf{H}_{dn}^{ss}&\mathbf{H}_{di}^{ss}\\ \mathbf{H}_{rd}^{ss}&\mathbf{H}_{rr}^{ss}&\mathbf{H}_{rn}^{ss}&\mathbf{H}_{ri}^{ss}\\ \mathbf{H}_{nd}^{ss}&\mathbf{H}_{nr}^{ss}&\mathbf{H}_{nn}^{ss}&\mathbf{H}_{ni}^{ss}\\ \mathbf{H}_{id}^{ss}&\mathbf{H}_{ir}^{ss}&\mathbf{H}_{in}^{ss}&\mathbf{H}_{ii}^{ss}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}_d^s\\ \widetilde{\mathbf{u}}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s\end{Bmatrix}+\begin{bmatrix}\mathbf{H}_{dd}^{sv}&\mathbf{H}_{dr}^{sv}&\mathbf{H}_{dn}^{sv}&\mathbf{H}_{di}^{sv}\\ \mathbf{H}_{rd}^{sv}&\mathbf{H}_{rr}^{sv}&\mathbf{H}_{rn}^{sv}&\mathbf{H}_{ri}^{sv}\\ \mathbf{H}_{nd}^{sv}&\mathbf{H}_{nr}^{sv}&\mathbf{H}_{nn}^{sv}&\mathbf{H}_{ni}^{sv}\\ \mathbf{H}_{id}^{sv}&\mathbf{H}_{ir}^{sv}&\mathbf{H}_{in}^{sv}&\mathbf{H}_{ii}^{sv}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}_d^v\\ \widetilde{\mathbf{u}}_r^v\\ \mathbf{u}_n^v\\ \mathbf{u}_i^v\end{Bmatrix}$$
$$=\begin{bmatrix}\mathbf{G}_{dd}^{ss}&\mathbf{G}_{dr}^{ss}&\mathbf{G}_{dn}^{ss}&\mathbf{G}_{di}^{ss}\\ \mathbf{G}_{rd}^{ss}&\mathbf{G}_{rr}^{ss}&\mathbf{G}_{rn}^{ss}&\mathbf{G}_{ri}^{ss}\\ \mathbf{G}_{nd}^{ss}&\mathbf{G}_{nr}^{ss}&\mathbf{G}_{nn}^{ss}&\mathbf{G}_{ni}^{ss}\\ \mathbf{G}_{id}^{ss}&\mathbf{G}_{ir}^{ss}&\mathbf{G}_{in}^{ss}&\mathbf{G}_{ii}^{ss}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \widetilde{\mathbf{q}}_r^s\\ \overline{\mathbf{q}}_n^s\\ \mathbf{q}_i^s\end{Bmatrix}+\begin{bmatrix}\mathbf{G}_{dd}^{sv}&\mathbf{G}_{dr}^{sv}&\mathbf{G}_{dn}^{sv}&\mathbf{G}_{di}^{sv}\\ \mathbf{G}_{rd}^{sv}&\mathbf{G}_{rr}^{sv}&\mathbf{G}_{rn}^{sv}&\mathbf{G}_{ri}^{sv}\\ \mathbf{G}_{nd}^{sv}&\mathbf{G}_{nr}^{sv}&\mathbf{G}_{nn}^{sv}&\mathbf{G}_{ni}^{sv}\\ \mathbf{G}_{id}^{sv}&\mathbf{G}_{ir}^{sv}&\mathbf{G}_{in}^{sv}&\mathbf{G}_{ii}^{sv}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^v\\ \widetilde{\mathbf{q}}_r^v\\ \overline{\mathbf{q}}_n^v\\ \mathbf{q}_i^v\end{Bmatrix} \tag{7}$$

where the subscripts 'd, n, r, i' represent the Dirichlet, Neumann, Robin and interface boundaries, respectively. The superscript '– ' indicates the physical quantities are known.

Using robin boundary conditions $\widetilde{\mathbf{q}}_r^v=\widetilde{\beta}_r^v-\widetilde{\alpha}_r^v\widetilde{\mathbf{u}}_r^v$ to eliminate the DOF of virtual nodes located on the robin boundaries. Eq. (7) becomes:

$$\begin{bmatrix}\mathbf{H}_{dd}^{ss}&\mathbf{H}_{dr}^{ss}&\mathbf{H}_{dn}^{ss}&\mathbf{H}_{di}^{ss}\\ \mathbf{H}_{rd}^{ss}&\mathbf{H}_{rr}^{ss}&\mathbf{H}_{rn}^{ss}&\mathbf{H}_{ri}^{ss}\\ \mathbf{H}_{nd}^{ss}&\mathbf{H}_{nr}^{ss}&\mathbf{H}_{nn}^{ss}&\mathbf{H}_{ni}^{ss}\\ \mathbf{H}_{id}^{ss}&\mathbf{H}_{ir}^{ss}&\mathbf{H}_{in}^{ss}&\mathbf{H}_{ii}^{ss}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}_d^s\\ \widetilde{\mathbf{u}}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s\end{Bmatrix}+\begin{bmatrix}\mathbf{H}_{dd}^{sv}&\mathbf{H}_{dr}^{sv}+\mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^v&\mathbf{H}_{dn}^{sv}&\mathbf{H}_{di}^{sv}\\ \mathbf{H}_{rd}^{sv}&\mathbf{H}_{rr}^{sv}+\mathbf{G}_{rr}^{sv}\widetilde{\alpha}_r^v&\mathbf{H}_{rn}^{sv}&\mathbf{H}_{ri}^{sv}\\ \mathbf{H}_{nd}^{sv}&\mathbf{H}_{nr}^{sv}+\mathbf{G}_{nr}^{sv}\widetilde{\alpha}_r^v&\mathbf{H}_{nn}^{sv}&\mathbf{H}_{ni}^{sv}\\ \mathbf{H}_{id}^{sv}&\mathbf{H}_{ir}^{sv}+\mathbf{G}_{ir}^{sv}\widetilde{\alpha}_r^v&\mathbf{H}_{in}^{sv}&\mathbf{H}_{ii}^{sv}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}_d^v\\ \widetilde{\mathbf{u}}_r^v\\ \mathbf{u}_n^v\\ \mathbf{u}_i^v\end{Bmatrix}$$
$$=\begin{bmatrix}\mathbf{G}_{dd}^{ss}&\mathbf{G}_{dr}^{ss}&\mathbf{G}_{dn}^{ss}&\mathbf{G}_{di}^{ss}\\ \mathbf{G}_{rd}^{ss}&\mathbf{G}_{rr}^{ss}&\mathbf{G}_{rn}^{ss}&\mathbf{G}_{ri}^{ss}\\ \mathbf{G}_{nd}^{ss}&\mathbf{G}_{nr}^{ss}&\mathbf{G}_{nn}^{ss}&\mathbf{G}_{ni}^{ss}\\ \mathbf{G}_{id}^{ss}&\mathbf{G}_{ir}^{ss}&\mathbf{G}_{in}^{ss}&\mathbf{G}_{ii}^{ss}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \widetilde{\mathbf{q}}_r^s\\ \overline{\mathbf{q}}_n^s\\ \mathbf{q}_i^s\end{Bmatrix}+\begin{bmatrix}\mathbf{G}_{dd}^{sv}&\mathbf{G}_{dr}^{sv}&\mathbf{G}_{dn}^{sv}&\mathbf{G}_{di}^{sv}\\ \mathbf{G}_{rd}^{sv}&\mathbf{G}_{rr}^{sv}&\mathbf{G}_{rn}^{sv}&\mathbf{G}_{ri}^{sv}\\ \mathbf{G}_{nd}^{sv}&\mathbf{G}_{nr}^{sc}&\mathbf{G}_{nn}^{sv}&\mathbf{G}_{ni}^{sv}\\ \mathbf{G}_{id}^{sv}&\mathbf{G}_{ir}^{sc}&\mathbf{G}_{in}^{sv}&\mathbf{G}_{ii}^{sv}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^v\\ \widetilde{\beta}_r^v\\ \overline{\mathbf{q}}_n^v\\ \mathbf{q}_i^v\end{Bmatrix} \tag{8}$$

Introducing the second-layer interpolation, the matrix forms of Eqs. (3,4) are given below:

$$\begin{Bmatrix}\overline{\mathbf{u}}_d^v\\ \widetilde{\mathbf{u}}_r^v\\ \mathbf{u}_n^v\\ \mathbf{u}_i^v\end{Bmatrix}=\begin{bmatrix}\mathbf{UU}_{dd}^{vs}&\mathbf{UU}_{dr}^{vs}&\mathbf{UU}_{dn}^{vs}&\mathbf{UU}_{di}^{vs}\\ \mathbf{UU}_{rd}^{vs}&\mathbf{UU}_{rr}^{vs}&\mathbf{UU}_{rn}^{vs}&\mathbf{UU}_{ri}^{vs}\\ \mathbf{UU}_{nd}^{vs}&\mathbf{UU}_{nr}^{vs}&\mathbf{UU}_{nn}^{vs}&\mathbf{UU}_{ni}^{vs}\\ \mathbf{UU}_{id}^{vs}&\mathbf{UU}_{ir}^{vs}&\mathbf{UU}_{in}^{vs}&\mathbf{UU}_{ii}^{vs}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}_d^s\\ \widetilde{\mathbf{u}}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s\end{Bmatrix}+\begin{bmatrix}\mathbf{UQ}_{dd}^{vs}&\mathbf{UQ}_{dr}^{vs}&\mathbf{UQ}_{dn}^{vs}&\mathbf{UQ}_{di}^{vs}\\ \mathbf{UQ}_{rd}^{vs}&\mathbf{UQ}_{rr}^{vs}&\mathbf{UQ}_{rn}^{vs}&\mathbf{UQ}_{ri}^{vs}\\ \mathbf{UQ}_{nd}^{vs}&\mathbf{UQ}_{nr}^{vs}&\mathbf{UQ}_{nn}^{vs}&\mathbf{UQ}_{ni}^{vs}\\ \mathbf{UQ}_{id}^{vs}&\mathbf{UQ}_{ir}^{vs}&\mathbf{UQ}_{in}^{vs}&\mathbf{UQ}_{i}^{vs}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \widetilde{\mathbf{q}}_r^s\\ \overline{\mathbf{q}}_n^s\\ \mathbf{q}_i^s\end{Bmatrix}, \tag{9}$$

$$\begin{Bmatrix}\mathbf{q}_d^v\\ \widetilde{\mathbf{q}}_r^v\\ \overline{\mathbf{q}}_n^v\\ \mathbf{q}_i^v\end{Bmatrix}=\begin{bmatrix}\mathbf{QU}_{dd}^{vs}&\mathbf{QU}_{dr}^{vs}&\mathbf{QU}_{dn}^{vs}&\mathbf{QU}_{di}^{vs}\\ \mathbf{QU}_{rd}^{vs}&\mathbf{QU}_{rr}^{vs}&\mathbf{QU}_{rn}^{vs}&\mathbf{QU}_{ri}^{vs}\\ \mathbf{QU}_{nd}^{vs}&\mathbf{QU}_{nr}^{vs}&\mathbf{QU}_{nn}^{vs}&\mathbf{QU}_{ni}^{vs}\\ \mathbf{QU}_{id}^{vs}&\mathbf{QU}_{ir}^{vs}&\mathbf{QU}_{in}^{vs}&\mathbf{QU}_{ii}^{vs}\end{bmatrix}\begin{Bmatrix}\overline{\mathbf{u}}_d^s\\ \widetilde{\mathbf{u}}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s\end{Bmatrix}+\begin{bmatrix}\mathbf{QQ}_{dd}^{vs}&\mathbf{QQ}_{dr}^{vs}&\mathbf{QQ}_{dn}^{vs}&\mathbf{QQ}_{di}^{vs}\\ \mathbf{QQ}_{rd}^{vs}&\mathbf{QQ}_{rr}^{vs}&\mathbf{QQ}_{rn}^{vs}&\mathbf{QQ}_{ri}^{vs}\\ \mathbf{QQ}_{nd}^{vs}&\mathbf{QQ}_{nr}^{vs}&\mathbf{QQ}_{nn}^{vs}&\mathbf{QQ}_{nn}^{vs}\\ \mathbf{QQ}_{id}^{vs}&\mathbf{QQ}_{ir}^{vs}&\mathbf{QQ}_{in}^{vs}&\mathbf{QQ}_{ii}^{vs}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \widetilde{\mathbf{q}}_r^s\\ \overline{\mathbf{q}}_n^s\\ \mathbf{q}_i^s\end{Bmatrix}, \tag{10}$$

where **UU, UQ, QU, QQ** represent the shape functions in Eqs. (3,4).

Substituting Eqs. (9,10) into the Eq. (8) and rearranging, we obtain:

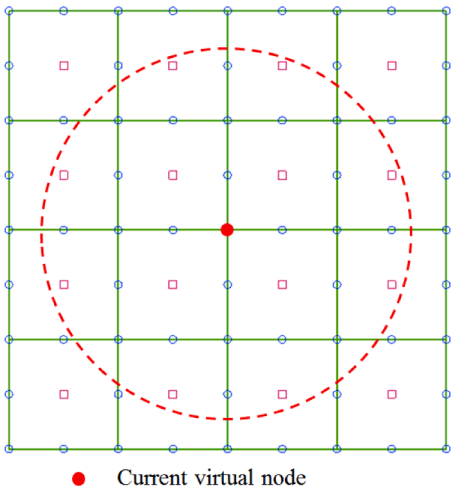

Fig. 3. Schematic diagram of the second-layer interpolation.

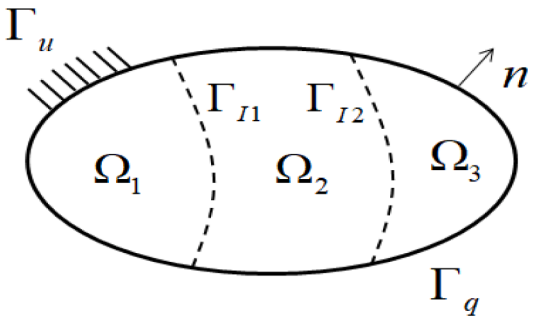

Fig. 4. Multi-domain potential problem.

$$
\begin{aligned}
&\begin{Bmatrix}
\mathbf{H}_{dd}^{ss} & \mathbf{H}_{dr}^{ss} & \mathbf{H}_{dn}^{ss} & \mathbf{H}_{di}^{ss}\\
\mathbf{H}_{rd}^{ss} & \mathbf{H}_{rr}^{ss} & \mathbf{H}_{rn}^{ss} & \mathbf{H}_{ri}^{ss}\\
\mathbf{H}_{nd}^{ss} & \mathbf{H}_{nr}^{ss} & \mathbf{H}_{nn}^{ss} & \mathbf{H}_{ni}^{ss}\\
\mathbf{H}_{id}^{ss} & \mathbf{H}_{ir}^{ss} & \mathbf{H}_{in}^{ss} & \mathbf{H}_{ii}^{ss}
\end{Bmatrix}
+\begin{bmatrix}
\mathbf{H}_{dr}^{sv}+\mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{dn}^{sv} & \mathbf{H}_{di}^{sv}\\
\mathbf{H}_{rr}^{sv}+\mathbf{G}_{rr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{rn}^{sv} & \mathbf{H}_{ri}^{sv}\\
\mathbf{H}_{nr}^{sv}+\mathbf{G}_{nr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{nn}^{sv} & \mathbf{H}_{ni}^{sv}\\
\mathbf{H}_{ir}^{sv}+\mathbf{G}_{ir}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{in}^{sv} & \mathbf{H}_{ii}^{sv}
\end{bmatrix}
\begin{bmatrix}
\mathbf{UU}_{rd}^{vs} & \mathbf{UU}_{rr}^{vs} & \mathbf{UU}_{rn}^{vs} & \mathbf{UU}_{ri}^{vs}\\
\mathbf{UU}_{nd}^{vs} & \mathbf{UU}_{nr}^{vs} & \mathbf{UU}_{nn}^{vs} & \mathbf{UU}_{ni}^{vs}\\
\mathbf{UU}_{id}^{vs} & \mathbf{UU}_{ir}^{vs} & \mathbf{UU}_{in}^{vs} & \mathbf{UU}_{ii}^{vs}
\end{bmatrix}
\begin{Bmatrix}
\overline{\mathbf{u}}_d^s\\ \widetilde{\mathbf{u}}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s
\end{Bmatrix}\\
&-\begin{bmatrix}
\mathbf{G}_{dd}^{sv} & \mathbf{G}_{di}^{sv}\\
\mathbf{G}_{rd}^{sv} & \mathbf{G}_{ri}^{sv}\\
\mathbf{G}_{nd}^{sv} & \mathbf{G}_{ni}^{sv}\\
\mathbf{G}_{id}^{sv} & \mathbf{G}_{ii}^{sv}
\end{bmatrix}
\begin{bmatrix}
\mathbf{QU}_{dd}^{vs} & \mathbf{QU}_{dr}^{vs} & \mathbf{QU}_{dn}^{vs} & \mathbf{QU}_{di}^{vs}\\
\mathbf{QU}_{id}^{vs} & \mathbf{QU}_{ir}^{vs} & \mathbf{QU}_{in}^{vs} & \mathbf{QU}_{ii}^{vs}
\end{bmatrix}
\begin{Bmatrix}
\overline{\mathbf{u}}_d^s\\ \widetilde{\mathbf{u}}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s
\end{Bmatrix}=\\
&\begin{Bmatrix}
\mathbf{G}_{dd}^{ss} & \mathbf{G}_{dr}^{ss} & \mathbf{G}_{dn}^{ss} & \mathbf{G}_{di}^{ss}\\
\mathbf{G}_{rd}^{ss} & \mathbf{G}_{rr}^{ss} & \mathbf{G}_{rn}^{ss} & \mathbf{G}_{ri}^{ss}\\
\mathbf{G}_{nd}^{ss} & \mathbf{G}_{nr}^{ss} & \mathbf{G}_{nn}^{ss} & \mathbf{G}_{ni}^{ss}\\
\mathbf{G}_{id}^{ss} & \mathbf{G}_{ir}^{ss} & \mathbf{G}_{in}^{ss} & \mathbf{G}_{ii}^{ss}
\end{Bmatrix}
+\begin{bmatrix}
\mathbf{G}_{dd}^{sv} & \mathbf{G}_{di}^{sv}\\
\mathbf{G}_{rd}^{sv} & \mathbf{G}_{ri}^{sv}\\
\mathbf{G}_{nd}^{sv} & \mathbf{G}_{ni}^{sv}\\
\mathbf{G}_{id}^{sv} & \mathbf{G}_{ii}^{sv}
\end{bmatrix}
\begin{bmatrix}
\mathbf{QQ}_{dd}^{vs} & \mathbf{QQ}_{dr}^{vs} & \mathbf{QQ}_{dn}^{vs} & \mathbf{QQ}_{di}^{vs}\\
\mathbf{QQ}_{id}^{vs} & \mathbf{QQ}_{ir}^{vs} & \mathbf{QQ}_{in}^{vs} & \mathbf{QQ}_{ii}^{vs}
\end{bmatrix}
\begin{Bmatrix}
\mathbf{q}_d^s\\ \widetilde{\mathbf{q}}_r^s\\ \overline{\mathbf{q}}_n^s\\ \mathbf{q}_i^s
\end{Bmatrix}\\
&-\begin{bmatrix}
\mathbf{H}_{dr}^{sv}+\mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{dn}^{sv} & \mathbf{H}_{di}^{sv}\\
\mathbf{H}_{rr}^{sv}+\mathbf{G}_{rr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{rn}^{sv} & \mathbf{H}_{ri}^{sv}\\
\mathbf{H}_{nr}^{sv}+\mathbf{G}_{nr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{nn}^{sv} & \mathbf{H}_{ni}^{sv}\\
\mathbf{H}_{ir}^{sv}+\mathbf{G}_{ir}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{in}^{sv} & \mathbf{H}_{ii}^{sv}
\end{bmatrix}
\begin{bmatrix}
\mathbf{UQ}_{rd}^{vs} & \mathbf{UQ}_{rr}^{vs} & \mathbf{UQ}_{rn}^{vs} & \mathbf{UQ}_{ri}^{vs}\\
\mathbf{UQ}_{nd}^{vs} & \mathbf{UQ}_{nr}^{vs} & \mathbf{UQ}_{nn}^{vs} & \mathbf{UQ}_{ni}^{vs}\\
\mathbf{UQ}_{id}^{vs} & \mathbf{UQ}_{ir}^{vs} & \mathbf{UQ}_{in}^{vs} & \mathbf{UQ}_{i}^{vs}
\end{bmatrix}
\begin{Bmatrix}
\mathbf{q}_d^s\\ \widetilde{\mathbf{q}}_r^s\\ \overline{\mathbf{q}}_n^s\\ \mathbf{q}_i^s
\end{Bmatrix}\\
&+\begin{bmatrix}
-\mathbf{H}_{dd}^{sv} & \mathbf{G}_{dr}^{sv} & \mathbf{G}_{dn}^{sv}\\
-\mathbf{H}_{rd}^{sv} & \mathbf{G}_{rr}^{sv} & \mathbf{G}_{rn}^{sv}\\
-\mathbf{H}_{nd}^{sv} & \mathbf{G}_{nr}^{sc} & \mathbf{G}_{nn}^{sv}\\
-\mathbf{H}_{id}^{sv} & \mathbf{G}_{ir}^{sc} & \mathbf{G}_{in}^{sv}
\end{bmatrix}
\begin{Bmatrix}
\overline{\mathbf{u}}_d^v\\ \widetilde{\beta}_r^v\\ \overline{\mathbf{q}}_n^v
\end{Bmatrix}
\end{aligned}
\tag{11}
$$

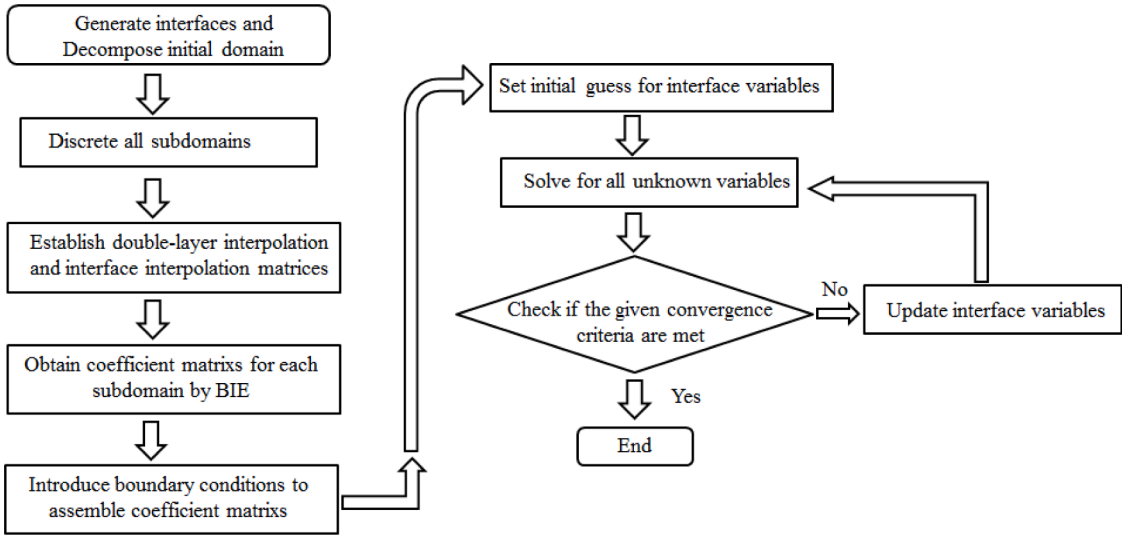

Fig. 5. The calculation flow chart of DDM.

where the blank positions of the above matrices indicate that all its elements are zero.

Considering robin boundary conditions of the source nodes $\widetilde{\mathbf{q}}_r^s = \widetilde{\beta}_r^s - \widetilde{\alpha}_r^s \widetilde{\mathbf{u}}_r^s$, Eq. (11) becomes:

$$
\begin{aligned}
&\left\{ \begin{bmatrix} \mathbf{H}_{dd}^{ss} & \mathbf{H}_{dr}^{ss} & \mathbf{H}_{dn}^{ss} & \mathbf{H}_{di}^{ss} \\ \mathbf{H}_{rd}^{ss} & \mathbf{H}_{rr}^{ss} & \mathbf{H}_{rn}^{ss} & \mathbf{H}_{ri}^{ss} \\ \mathbf{H}_{nd}^{ss} & \mathbf{H}_{nr}^{ss} & \mathbf{H}_{nn}^{ss} & \mathbf{H}_{ni}^{ss} \\ \mathbf{H}_{id}^{ss} & \mathbf{H}_{ir}^{ss} & \mathbf{H}_{in}^{ss} & \mathbf{H}_{ii}^{ss} \end{bmatrix} \right. \\
&+ \left[\begin{matrix} \mathbf{H}_{dr}^{sv} + \mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{dn}^{sv} & \mathbf{H}_{di}^{sv} \\ \mathbf{H}_{rr}^{sv} + \mathbf{G}_{rr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{rn}^{sv} & \mathbf{H}_{ri}^{sv} \\ \mathbf{H}_{nr}^{sv} + \mathbf{G}_{nr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{nn}^{sv} & \mathbf{H}_{ni}^{sv} \\ \mathbf{H}_{ir}^{sv} + \mathbf{G}_{ir}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{in}^{sv} & \mathbf{H}_{ii}^{sv} \end{matrix} \right. \left. \left[\begin{matrix} & & & \\ \mathbf{UU}_{rd}^{vs} & \mathbf{UU}_{rr}^{vs} & \mathbf{UU}_{rn}^{vs} & \mathbf{UU}_{ri}^{vs} \\ \mathbf{UU}_{nd}^{vs} & \mathbf{UU}_{nr}^{vs} & \mathbf{UU}_{nn}^{vs} & \mathbf{UU}_{ni}^{vs} \\ \mathbf{UU}_{id}^{vs} & \mathbf{UU}_{ir}^{vs} & \mathbf{UU}_{in}^{vs} & \mathbf{UU}_{ii}^{vs} \end{matrix} \right] \right\} \begin{Bmatrix} \overline{\mathbf{u}}_d^s \\ \widetilde{\mathbf{u}}_r^s \\ \mathbf{u}_n^s \\ \mathbf{u}_i^s \end{Bmatrix} \\
&+ \left\{ - \begin{bmatrix} \mathbf{G}_{dd}^{sv} & \mathbf{G}_{di}^{sv} \\ \mathbf{G}_{rd}^{sv} & \mathbf{G}_{ri}^{sv} \\ \mathbf{G}_{nd}^{sv} & \mathbf{G}_{ni}^{sv} \\ \mathbf{G}_{id}^{sv} & \mathbf{G}_{ii}^{sv} \end{bmatrix} \begin{bmatrix} \mathbf{QU}_{dd}^{vs} & \mathbf{QU}_{dr}^{vs} & \mathbf{QU}_{dn}^{vs} & \mathbf{QU}_{di}^{vs} \\ & & & \\ \mathbf{QU}_{id}^{vs} & \mathbf{QU}_{ir}^{vs} & \mathbf{QU}_{in}^{vs} & \mathbf{QU}_{ii}^{vs} \end{bmatrix} \right. \\
&\left. + \begin{bmatrix} \mathbf{G}_{dr}^{ss}\widetilde{\alpha}_r^s & & & \\ \mathbf{G}_{rr}^{ss}\widetilde{\alpha}_r^s & & & \\ \mathbf{G}_{nr}^{ss}\widetilde{\alpha}_r^s & & & \\ \mathbf{G}_{ir}^{ss}\widetilde{\alpha}_r^s & & & \end{bmatrix} \right\} \begin{Bmatrix} \overline{\mathbf{u}}_d^s \\ \widetilde{\mathbf{u}}_r^s \\ \mathbf{u}_n^s \\ \mathbf{u}_i^s \end{Bmatrix} \\
&+ \left\{ \begin{bmatrix} \mathbf{G}_{dd}^{sv} & \mathbf{G}_{di}^{sv} \\ \mathbf{G}_{rd}^{sv} & \mathbf{G}_{ri}^{sv} \\ \mathbf{G}_{nd}^{sv} & \mathbf{G}_{ni}^{sv} \\ \mathbf{G}_{id}^{sv} & \mathbf{G}_{ii}^{sv} \end{bmatrix} \begin{bmatrix} \mathbf{QQ}_{dr}^{vs}\widetilde{\alpha}_r^s \\ \\ \mathbf{QQ}_{ir}^{vs}\widetilde{\alpha}_r^s \end{bmatrix} \right. \\
&\left. - \begin{bmatrix} \mathbf{H}_{dr}^{sv} + \mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{dn}^{sv} & \mathbf{H}_{di}^{sv} \\ \mathbf{H}_{rr}^{sv} + \mathbf{G}_{rr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{rn}^{sv} & \mathbf{H}_{ri}^{sv} \\ \mathbf{H}_{nr}^{sv} + \mathbf{G}_{nr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{nn}^{sv} & \mathbf{H}_{ni}^{sv} \\ \mathbf{H}_{ir}^{sv} + \mathbf{G}_{ir}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{in}^{sv} & \mathbf{H}_{ii}^{sv} \end{bmatrix} \begin{bmatrix} \\ \mathbf{UQ}_{rr}^{vs}\widetilde{\alpha}_r^s \\ \mathbf{UQ}_{nr}^{vs}\widetilde{\alpha}_r^s \\ \mathbf{UQ}_{ir}^{vs}\widetilde{\alpha}_r^s \end{bmatrix} \right\} \begin{Bmatrix} \overline{\mathbf{u}}_d^s \\ \widetilde{\mathbf{u}}_r^s \\ \mathbf{u}_n^s \\ \mathbf{u}_i^s \end{Bmatrix} = \\
&\left(\begin{bmatrix} \mathbf{G}_{dd}^{ss} & \mathbf{G}_{dr}^{ss} & \mathbf{G}_{dn}^{ss} & \mathbf{G}_{di}^{ss} \\ \mathbf{G}_{rd}^{ss} & \mathbf{G}_{rr}^{ss} & \mathbf{G}_{rn}^{ss} & \mathbf{G}_{ri}^{ss} \\ \mathbf{G}_{nd}^{ss} & \mathbf{G}_{nr}^{ss} & \mathbf{G}_{nn}^{ss} & \mathbf{G}_{ni}^{ss} \\ \mathbf{G}_{id}^{ss} & \mathbf{G}_{ir}^{ss} & \mathbf{G}_{in}^{ss} & \mathbf{G}_{ii}^{ss} \end{bmatrix} \right. \\
&\left. + \begin{bmatrix} \mathbf{G}_{dd}^{sv} & \mathbf{G}_{di}^{sv} \\ \mathbf{G}_{rd}^{sv} & \mathbf{G}_{ri}^{sv} \\ \mathbf{G}_{nd}^{sv} & \mathbf{G}_{ni}^{sv} \\ \mathbf{G}_{id}^{sv} & \mathbf{G}_{ii}^{sv} \end{bmatrix} \begin{bmatrix} \mathbf{QQ}_{dd}^{vs} & \mathbf{QQ}_{dr}^{vs} & \mathbf{QQ}_{dn}^{vs} & \mathbf{QQ}_{di}^{vs} \\ & & & \\ \mathbf{QQ}_{id}^{vs} & \mathbf{QQ}_{ir}^{vs} & \mathbf{QQ}_{in}^{vs} & \mathbf{QQ}_{ii}^{vs} \end{bmatrix} \right) \begin{Bmatrix} \mathbf{q}_d^s \\ \widetilde{\beta}_r^s \\ \overline{\mathbf{q}}_n^s \\ \mathbf{q}_i^s \end{Bmatrix} \\
&- \left\{ \begin{bmatrix} \mathbf{H}_{dr}^{sv} + \mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{dn}^{sv} & \mathbf{H}_{di}^{sv} \\ \mathbf{H}_{rr}^{sv} + \mathbf{G}_{rr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{rn}^{sv} & \mathbf{H}_{ri}^{sv} \\ \mathbf{H}_{nr}^{sv} + \mathbf{G}_{nr}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{nn}^{sv} & \mathbf{H}_{ni}^{sv} \\ \mathbf{H}_{ir}^{sv} + \mathbf{G}_{ir}^{sv}\widetilde{\alpha}_r^v & \mathbf{H}_{in}^{sv} & \mathbf{H}_{ii}^{sv} \end{bmatrix} \begin{bmatrix} & & & \\ \mathbf{UQ}_{rd}^{vs} & \mathbf{UQ}_{rr}^{vs} & \mathbf{UQ}_{rn}^{vs} & \mathbf{UQ}_{ri}^{vs} \\ \mathbf{UQ}_{nd}^{vs} & \mathbf{UQ}_{nr}^{vs} & \mathbf{UQ}_{nn}^{vs} & \mathbf{UQ}_{ni}^{vs} \\ \mathbf{UQ}_{id}^{vs} & \mathbf{UQ}_{ir}^{vs} & \mathbf{UQ}_{in}^{vs} & \mathbf{UQ}_{i}^{vs} \end{bmatrix} \right\} \begin{Bmatrix} \mathbf{q}_d^s \\ \widetilde{\beta}_r^s \\ \overline{\mathbf{q}}_n^s \\ \mathbf{q}_i^s \end{Bmatrix} \begin{bmatrix} -\mathbf{H}_{dd}^{sv} & \mathbf{G}_{dr}^{sv} & \mathbf{G}_{dn}^{sv} \\ -\mathbf{H}_{rd}^{sv} & \mathbf{G}_{rr}^{sv} & \mathbf{G}_{rn}^{sv} \\ -\mathbf{H}_{nd}^{sv} & \mathbf{G}_{nr}^{sc} & \mathbf{G}_{nn}^{sv} \\ -\mathbf{H}_{id}^{sv} & \mathbf{G}_{ir}^{sc} & \mathbf{G}_{in}^{sv} \end{bmatrix} \begin{Bmatrix} \overline{\mathbf{u}}_d^v \\ \widetilde{\beta}_r^v \\ \overline{\mathbf{q}}_n^v \end{Bmatrix}
\end{aligned} \tag{12}
$$

With relocation and simplification, Eq. (12) can be written as follows:

$$
\begin{bmatrix} \mathbf{A}_{dr} & \mathbf{A}_{dn} & \mathbf{B}_{di} \\ \mathbf{A}_{rr} & \mathbf{A}_{rn} & \mathbf{B}_{ri} \\ \mathbf{A}_{nr} & \mathbf{A}_{nn} & \mathbf{B}_{ni} \\ \mathbf{A}_{ir} & \mathbf{A}_{in} & \mathbf{B}_{ii} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_r^s \\ \mathbf{u}_n^s \\ \mathbf{u}_i^s \end{Bmatrix} + \begin{bmatrix} \mathbf{A}_{dd} & \mathbf{A}_{di} \\ \mathbf{A}_{rd} & \mathbf{A}_{ri} \\ \mathbf{A}_{nd} & \mathbf{A}_{ni} \\ \mathbf{A}_{id} & \mathbf{A}_{ii} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_d^s \\ \\ \mathbf{q}_i^s \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_d \\ \mathbf{R}_r \\ \mathbf{R}_n \\ \mathbf{R}_i \end{Bmatrix} \tag{13}
$$

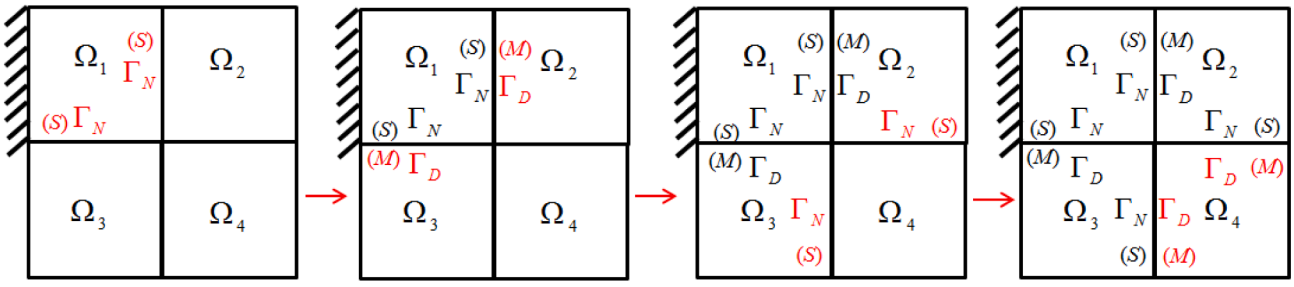

Fig. 6. The schematic diagram of interface classification scheme.

where **A** and **B** are the coefficient matrices and **R**represents the right-hand vector that has not yet considered the interface conditions. Taking $\mathbf{A}_{dd}$,$\mathbf{B}_{di}$and $\mathbf{R}_d$ as examples, their forms are given below:

$$\mathbf{A}_{dd} = -\left(\mathbf{G}_{dd}^{ss} + \mathbf{G}_{dd}^{sv}\mathbf{QQ}_{dd}^{vs} + \mathbf{G}_{di}^{sv}\mathbf{QQ}_{id}^{vs}\right) + \left(\mathbf{H}_{dr}^{sv} + \mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^{v}\right)\mathbf{UQ}_{rd}^{vs} + \mathbf{H}_{dn}^{sv}\mathbf{UQ}_{nd}^{vs} + \mathbf{H}_{di}^{sv}\mathbf{UQ}_{id}^{vs} \tag{14}$$

$$\mathbf{B}_{di} = \mathbf{H}_{di}^{ss} + \left(\mathbf{H}_{dr}^{sv} + \mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^{v}\right)\mathbf{UU}_{ri}^{vs} + \mathbf{H}_{dn}^{sv}\mathbf{UU}_{ni}^{vs} + \mathbf{H}_{di}^{sv}\mathbf{UU}_{ii}^{vs} - \left(\mathbf{G}_{dd}^{sv}\mathbf{QU}_{di}^{vs} + \mathbf{G}_{di}^{sv}\mathbf{QU}_{ii}^{vs}\right) \tag{15}$$

$$\begin{aligned}\mathbf{R}_d = {} & \left(-\mathbf{H}_{dd}^{ss}\overline{\mathbf{u}}_d^{s} + \mathbf{G}_{dr}^{ss}\widetilde{\beta}_r^{s} + \mathbf{G}_{dn}^{ss}\overline{\mathbf{q}}_n^{s}\right) + \left(\mathbf{G}_{dr}^{sv}\widetilde{\beta}_r^{v} + \mathbf{G}_{dn}^{sv}\overline{\mathbf{q}}_n^{v}\right) + \mathbf{G}_{dd}^{sv}\left(\mathbf{QQ}_{dr}^{vs}\widetilde{\beta}_r^{s} + \mathbf{QQ}_{dn}^{vs}\overline{\mathbf{q}}_n^{s} + \mathbf{QU}_{dd}^{vs}\overline{\mathbf{u}}_d^{s}\right)\\ & + \mathbf{G}_{di}^{sv}\left(\mathbf{QQ}_{ir}^{vs}\widetilde{\beta}_r^{s} + \mathbf{QQ}_{in}^{vs}\overline{\mathbf{q}}_n^{s} + \mathbf{QU}_{id}^{vs}\overline{\mathbf{u}}_d^{s}\right) - \left(\mathbf{H}_{dr}^{sv} + \mathbf{G}_{dr}^{sv}\widetilde{\alpha}_r^{v}\right)\left(\mathbf{UQ}_{rr}^{vs}\widetilde{\beta}_r^{s} + \mathbf{UQ}_{rn}^{vs}\overline{\mathbf{q}}_n^{s} + \mathbf{UU}_{rd}^{vs}\overline{\mathbf{u}}_d^{s}\right)\\ & - \mathbf{H}_{dn}^{sv}\left(\mathbf{UQ}_{nr}^{vs}\widetilde{\beta}_r^{s} + \mathbf{UQ}_{nn}^{vs}\overline{\mathbf{q}}_n^{s} + \mathbf{UU}_{nd}^{vs}\overline{\mathbf{u}}_d^{s}\right) - \mathbf{H}_{di}^{sv}\left(\mathbf{UQ}_{ir}^{vs}\widetilde{\beta}_r^{s} + \mathbf{UQ}_{in}^{vs}\overline{\mathbf{q}}_n^{s} + \mathbf{UU}_{id}^{vs}\overline{\mathbf{u}}_d^{s}\right) - \mathbf{H}_{dd}^{sv}\overline{\mathbf{u}}_d^{v}\end{aligned} \tag{16}$$

For interface boundaries, there are three available boundary conditions. Firstly, considering the application of Dirichlet boundary conditions on the interfaces, i.e., $\mathbf{u}_i^s = \overline{\mathbf{u}}_i^s$, Eq. (13) becomes

$$\begin{bmatrix}\mathbf{A}_{dd} & \mathbf{A}_{dr} & \mathbf{A}_{dn} & \mathbf{A}_{di}\\ \mathbf{A}_{rd} & \mathbf{A}_{rr} & \mathbf{A}_{rn} & \mathbf{A}_{ri}\\ \mathbf{A}_{nd} & \mathbf{A}_{nr} & \mathbf{A}_{nn} & \mathbf{A}_{ni}\\ \mathbf{A}_{id} & \mathbf{A}_{ir} & \mathbf{A}_{in} & \mathbf{A}_{ii}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \mathbf{u}_r^s\\ \mathbf{u}_n^s\\ \mathbf{q}_i^s\end{Bmatrix} = \begin{Bmatrix}\mathbf{R}_d - \mathbf{B}_{di}\overline{\mathbf{u}}_i^s\\ \mathbf{R}_r - \mathbf{B}_{ri}\overline{\mathbf{u}}_i^s\\ \mathbf{R}_n - \mathbf{B}_{ni}\overline{\mathbf{u}}_i^s\\ \mathbf{R}_i - \mathbf{B}_{ii}\overline{\mathbf{u}}_i^s\end{Bmatrix} \tag{17}$$

Then considering Neumann boundary conditions $\mathbf{q}_i^s = \overline{\mathbf{q}}_i^s$, we obtain

$$\begin{bmatrix}\mathbf{A}_{dd} & \mathbf{A}_{dr} & \mathbf{A}_{dn} & \mathbf{B}_{di}\\ \mathbf{A}_{rd} & \mathbf{A}_{rr} & \mathbf{A}_{rn} & \mathbf{B}_{ri}\\ \mathbf{A}_{nd} & \mathbf{A}_{nr} & \mathbf{A}_{nn} & \mathbf{B}_{ni}\\ \mathbf{A}_{id} & \mathbf{A}_{ir} & \mathbf{A}_{in} & \mathbf{B}_{ii}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \mathbf{u}_r^s\\ \mathbf{u}_n^s\\ \mathbf{u}_i^s\end{Bmatrix} = \begin{Bmatrix}\mathbf{R}_d - \mathbf{A}_{di}\overline{\mathbf{q}}_i^s\\ \mathbf{R}_r - \mathbf{A}_{ri}\overline{\mathbf{q}}_i^s\\ \mathbf{R}_n - \mathbf{A}_{ni}\overline{\mathbf{q}}_i^s\\ \mathbf{R}_i - \mathbf{A}_{ii}\overline{\mathbf{q}}_i^s\end{Bmatrix} \tag{18}$$

The last is Robin boundary conditions $\mathbf{u}_i^s = \boldsymbol{\alpha} - \beta\mathbf{q}_i^s$, we have

$$\begin{bmatrix}\mathbf{A}_{dd} & \mathbf{A}_{dr} & \mathbf{A}_{dn} & \mathbf{A}_{di} - \beta\mathbf{B}_{di}\\ \mathbf{A}_{rd} & \mathbf{A}_{rr} & \mathbf{A}_{rn} & \mathbf{A}_{ri} - \beta\mathbf{B}_{ri}\\ \mathbf{A}_{nd} & \mathbf{A}_{nr} & \mathbf{A}_{nn} & \mathbf{A}_{ni} - \beta\mathbf{B}_{ni}\\ \mathbf{A}_{id} & \mathbf{A}_{ir} & \mathbf{A}_{in} & \mathbf{A}_{ii} - \beta\mathbf{B}_{ii}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_d^s\\ \mathbf{u}_r^s\\ \mathbf{u}_n^s\\ \mathbf{q}_i^s\end{Bmatrix} = \begin{Bmatrix}\mathbf{R}_d - \mathbf{B}_{di}\boldsymbol{\alpha}\\ \mathbf{R}_r - \mathbf{B}_{ri}\boldsymbol{\alpha}\\ \mathbf{R}_n - \mathbf{B}_{ni}\boldsymbol{\alpha}\\ \mathbf{R}_i - \mathbf{B}_{ii}\boldsymbol{\alpha}\end{Bmatrix} \tag{19}$$

where $\boldsymbol{\alpha}$ is a vector with constant robin coefficient and β is the other constant robin coefficient.

Eqs. (17–19) can both be written in the form of System of linear equations $\mathbf{Ax} = \mathbf{b}$, and each subdomain can be solved separately. In addition, the coefficient matrix is invariant and only the values of the nodes on the interface boundaries need to be updated in each iteration step.

3. Five non-overlapping DDMs-DiBFM

In the implementation of non-overlapping DDMs, the boundary conditions on the interfaces are assumed and depended on the DDM used. The non-overlapping DDMs using Dirichlet-Dirichlet, Dirichlet-Neumann, Neumann-Neumann, Robin-Robin interface conditions are discussed in this section. The symbols '*M*' and '*S*' are used for distinguishing the master and slave interfaces belonging to different subdomains. If the interface conditions are of the same type, the master and slave interfaces can be specified arbitrarily. Otherwise, the interface with constrain condition is designated as the master interface. In each iteration step, the values of the nodes on the master interface are updated by iteration equation, while those on the slave interface are updated by iterative equation or interpolation between the two interfaces according to the DDM used. The calculation flow chart of DDM is shown below (Fig. 5).

3.1. Parallel Dirichlet-Dirichlet case

In this case, the Dirichlet boundary conditions are applied to all interfaces. The computations for all subdomains are carried out in parallel. This algorithm can be described below:

1. Set initial values for all interfaces, in general, $\mathbf{u}_{i,M}^1 = \mathbf{u}_{i,S}^1 = \mathbf{0}$.
2. Solve all subdomains in parallel and obtain the fluxes $\mathbf{q}_{i,M}^{nItr}$ and $\mathbf{q}_{i,S}^{nItr}$.
3. Update nodal values: $\mathbf{u}_{i,M}^{nItr+1} = \mathbf{u}_{i,M}^{nItr} - \alpha(\mathbf{q}_{i,M}^{nItr} + \mathbf{q}_{i,S}^{nItr})$, $\mathbf{u}_{i,S}^{nItr+1} = \mathbf{u}_{i,M}^{nItr+1}$. Determine whether the convergence criterion $\| \mathbf{u}_{i,M}^{nItr} - \mathbf{u}_{i,M}^{nItr-1} \| < \varepsilon$ is satisfied.
4. Repeat steps 2–3 until the convergence criterion is met.

where *nItr* and *i* represent the *n*-th iteration step and the *i* th interface respectively.αis a relaxation parameter and ε is the given tolerance.

3.2. Parallel Neumann - Neumann case

In the implementation of this algorithm, the Neumann boundary conditions are imposed on all interfaces instead of the Dirichlet boundary conditions, the computations are also parallel. The algorithm is given as follows:

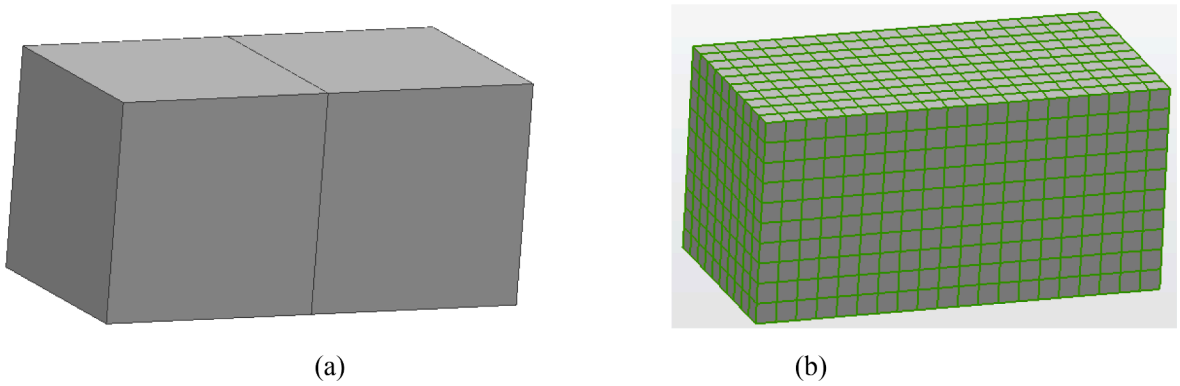

Fig. 7. (a) Cuboid and (b) its binary tree meshes.

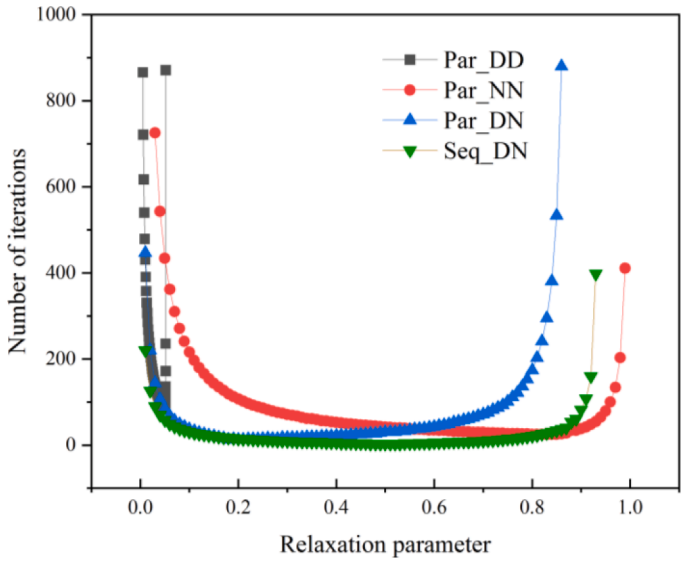

Fig. 8. Comparison of iterations under different relaxation parameters.

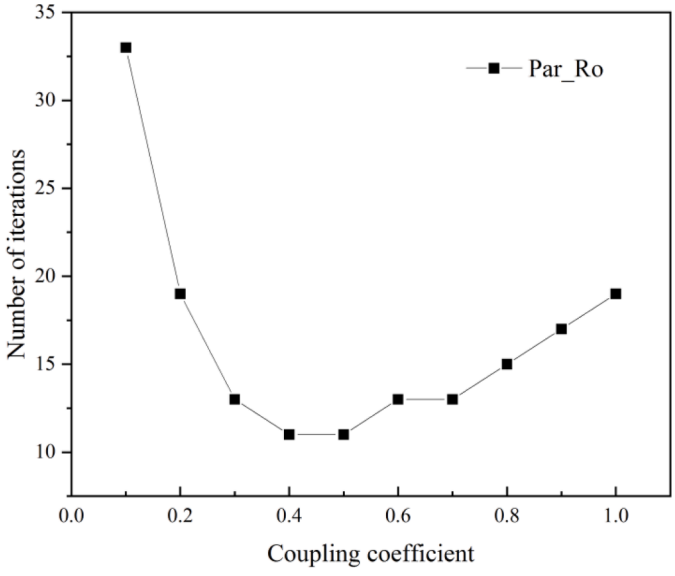

Fig. 9. Comparison of iterations under different coupling coefficients.

Table 2
Comparison of iterations and errors between optimal and dynamic relaxation parameter.

Algorithm	Optimal			Dynamic		
	nItrs	*Err_u*	*Err_q*	nItrs	*Err_u*	*Err_q*
Par_DD	83	1.321E-03	3.865E-03	29	1.305E-03	3.857E-03
Par_NN	25	1.313E-03	3.843E-03	11	1.309E-03	3.881E-03
Par_DN	15	1.309E-03	3.862E-03	17	1.305E-03	3.869E-03
Seq_DN	3	1.327E-03	3.890E-03	3	1.282E-03	3.781E-03

1. Set initial values for all interfaces, $\mathbf{q}_{i,M}^{1} = \mathbf{q}_{i,S}^{1} = \mathbf{0}$.
2. Solve all subdomains in parallel and obtain the potentials $\mathbf{u}_{i,M}^{nItr}$ and $\mathbf{u}_{i,S}^{nItr}$.
3. Check for the convergence criterion $\| \mathbf{u}_{i,M}^{nItr} - \mathbf{u}_{i,S}^{nItr} \| < \varepsilon$. If the convergence is not achieved, then update nodal values: $\mathbf{q}_{i,M}^{nItr+1} = \mathbf{q}_{i,M}^{nItr} + \beta(\mathbf{u}_{i,S}^{nItr} - \mathbf{u}_{i,M}^{nItr})$, $\mathbf{q}_{i,S}^{nItr+1} = -\mathbf{q}_{i,M}^{nItr+1}$.
4. Repeat steps 2–3 until the convergence criterion is met.

where$\beta$is a relaxation parameter.

This algorithm has its limitation, that is, Neumann boundary conditions can't be imposed to all outer boundaries of each subdomain. Otherwise, the coefficient matrix is singular and the solution is incorrect.

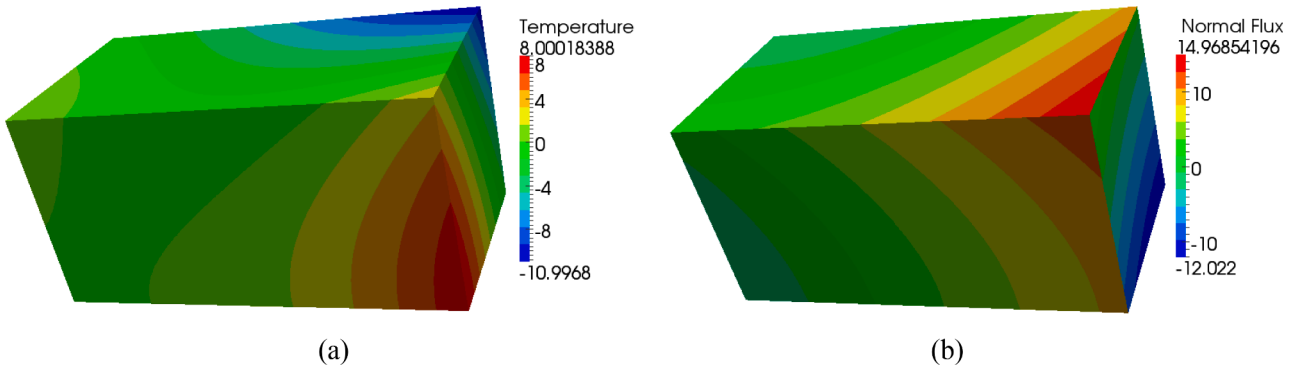

Fig. 10. Contours of (a) temperature and (b) normal flux of the cuboid.

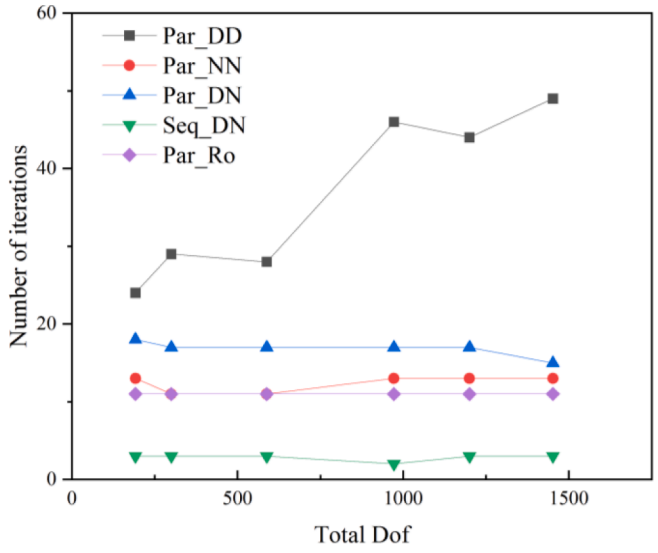

Fig. 11. Comparison of iterations under different mesh sizes.

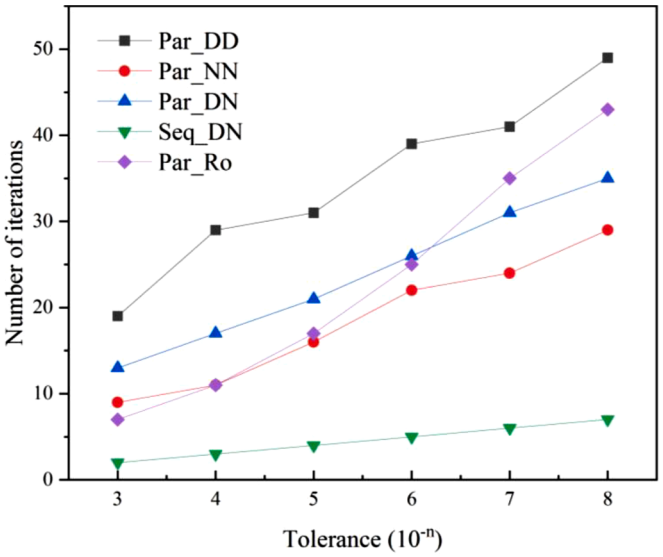

Fig. 12. Comparison of iterations under different tolerances.

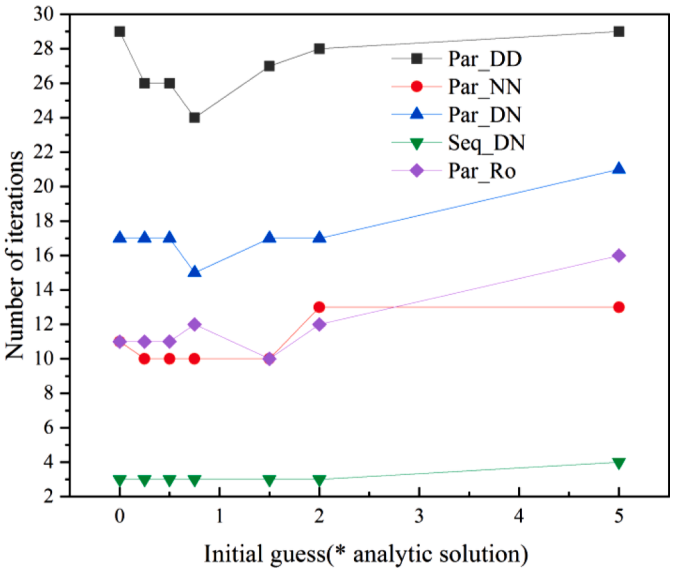

Fig. 13. Comparison of iterations under different initial guesses.

3.3. Sequential Dirichlet–Neumann case

In this case, the Dirichlet boundary conditions are applied to the master interfaces while the Neumann boundary conditions are applied to the slave interfaces, and the solving process is sequential. This algorithm is given as follows:

1. Set initial values for all master interfaces, $\mathbf{u}_{i,M}^{1} = \mathbf{0}$.
2. Solve the subdomains containing only the master interfaces and obtain the fluxes $\mathbf{q}_{i,M}^{nItr}$.
3. Set $\mathbf{q}_{i,S}^{nItr} = -\mathbf{q}_{i,M}^{nItr}$, Solve the subdomains containing the slave interfaces and obtain the potentials $\mathbf{u}_{i,S}^{nItr}$.
4. Determine whether the convergence criterion $\| \mathbf{u}_{i,M}^{nItr} - \mathbf{u}_{i,M}^{nItr-1} \| < \varepsilon$ is satisfied. If not, then update nodal values: $\mathbf{u}_{i,M}^{nItr+1} = \gamma \mathbf{u}_{i,S}^{nItr} + (1-\gamma)\mathbf{u}_{i,M}^{nItr}$.
5. Repeat steps 2–4 until the convergence criterion is satisfied.

where γis a relaxation parameter.

To ensure that each subdomain is constrained, there must be a master interface for a subdomain whose surface boundary conditions are all Neumann. As shown in Fig. 6, the interface classification scheme is as follows:

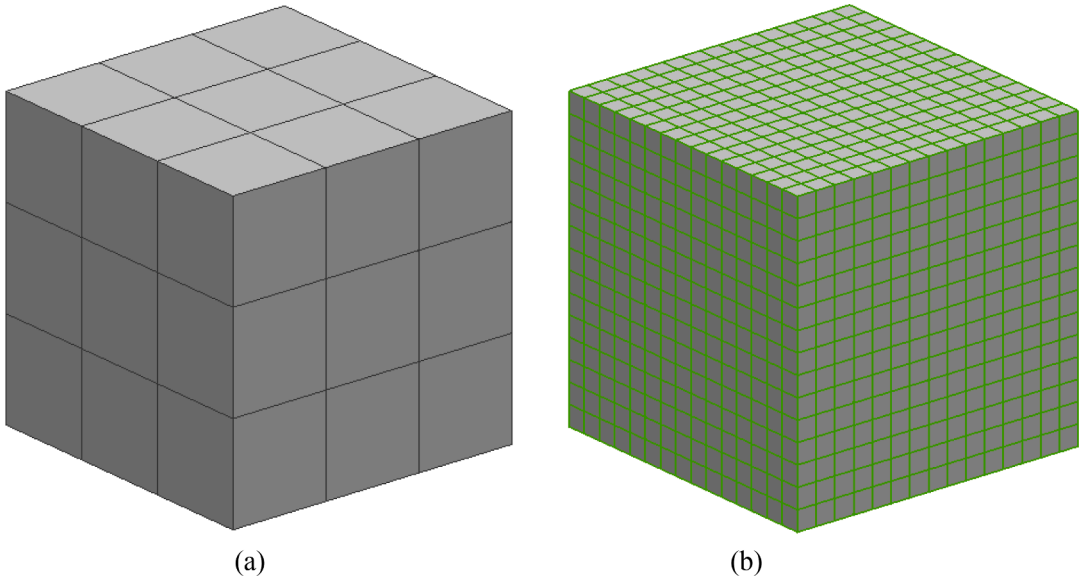

Fig. 14. (a) Cube with 27 subdomains and (b) its binary tree meshes.

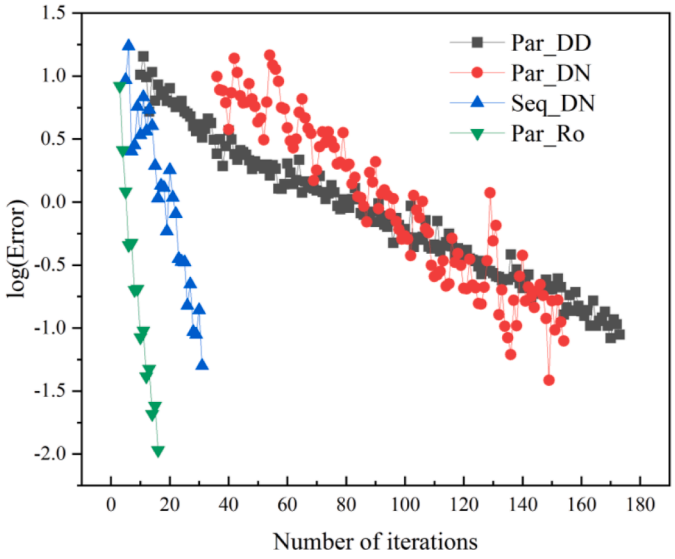

Fig. 15. The maximum interface error versus the number of iterations.

Table 3
Comparison of iterations and errors between four DiBFM-DDMs.

Algorithm	nItrs	*Err_u*	*Err_q*
Par_DD	440	7.2E-5	7.188E-4
Par_DN	313	7.09E-5	7.176E-4
Seq_DN	79	7.06E-5	7.085E-4
Par_Ro	33	7.11E-5	7.202E-4

1. Find subdomains containing constraints and mark all their interfaces as slave interfaces.
2. Mark another interface corresponding to the slave interface as the master interface.
3. The subdomains to which the master interfaces belong are the new constraint domains, then repeat step 1–2, skip if the interface is already marked.

Γ_D and $\Gamma_N$represent interfaces where the Dirichlet and Neumann boundary conditions are applied, respectively. This process is recursive and stops when all interfaces are marked.

3.4. Parallel Dirichlet–Neumann case

The application of boundary conditions and interface classification for this algorithm are the same as in 3.3, but the solving process is parallel. This algorithm is described below:

1. Set initial values for all interfaces, $\mathbf{u}_{i,M}^{1} = \mathbf{0}$, $\mathbf{q}_{i,S}^{1} = \mathbf{0}$.
2. Solve all subdomains in parallel and obtain the potentials $\mathbf{u}_{i,S}^{nItr}$ and the fluxes $\mathbf{q}_{i,M}^{nItr}$.
3. Check for the convergence criterion $\| \mathbf{u}_{i,M}^{nItr} - \mathbf{u}_{i,M}^{nItr-1} \| + \| \mathbf{q}_{i,S}^{nItr} - \mathbf{q}_{i,S}^{nItr-1} \| < \varepsilon$. If the convergence is not achieved, then update nodal values: $\mathbf{u}_{i,M}^{nItr+1} = \rho \mathbf{u}_{i,S}^{nItr} + (1-\rho)\mathbf{u}_{i,M}^{nItr}$, $\mathbf{q}_{i,S}^{nItr+1} = -\mathbf{q}_{i,M}^{nItr+1}$.
4. Repeat steps 2–3 until the convergence criterion is met.

where ρis a relaxation parameter.

3.5. Parallel Robin- Robin case

In this scheme, the Robin boundary conditions are applied to all interfaces, the computations for all subdomains are carried out in parallel. This algorithm can be described below:

1. Set initial values for all interfaces, $\mathbf{u}_{i,M}^{1} = \mathbf{u}_{i,S}^{1} = \mathbf{0}$.
2. Solve all subdomains in parallel and obtain the fluxes $\mathbf{q}_{i,M}^{nItr}$ and $\mathbf{q}_{i,S}^{nItr}$.
3. Construct the Robin boundary conditions $\mathbf{u}_{i,M}^{nItr+1} + \lambda \mathbf{q}_{i,M}^{nItr+1} = \mathbf{u}_{i,S}^{nItr} - \lambda \mathbf{q}_{i,S}^{nItr}$ and $\mathbf{u}_{i,S}^{nItr+1} + \lambda \mathbf{q}_{i,S}^{nItr+1} = \mathbf{u}_{i,M}^{nItr} - \lambda \mathbf{q}_{i,M}^{nItr}$.
4. Solve all subdomains in parallel. Obtain the potentials $\mathbf{u}_{i,M}^{nItr}, \mathbf{u}_{i,S}^{nItr}$ and the fluxes $\mathbf{q}_{i,M}^{nItr}, \mathbf{q}_{i,S}^{nItr}$.
5. Determine whether the convergence criterion $\| (\mathbf{u}_{i,M}^{nItr} + \lambda \mathbf{q}_{i,M}^{nItr}) - (\mathbf{u}_{i,M}^{nItr-1} + \lambda \mathbf{q}_{i,M}^{nItr-1}) \| < \varepsilon$ is satisfied.
6. Repeat steps 3–5 until the convergence criterion is met.

where $\lambda > 0$is the coupling coefficient.

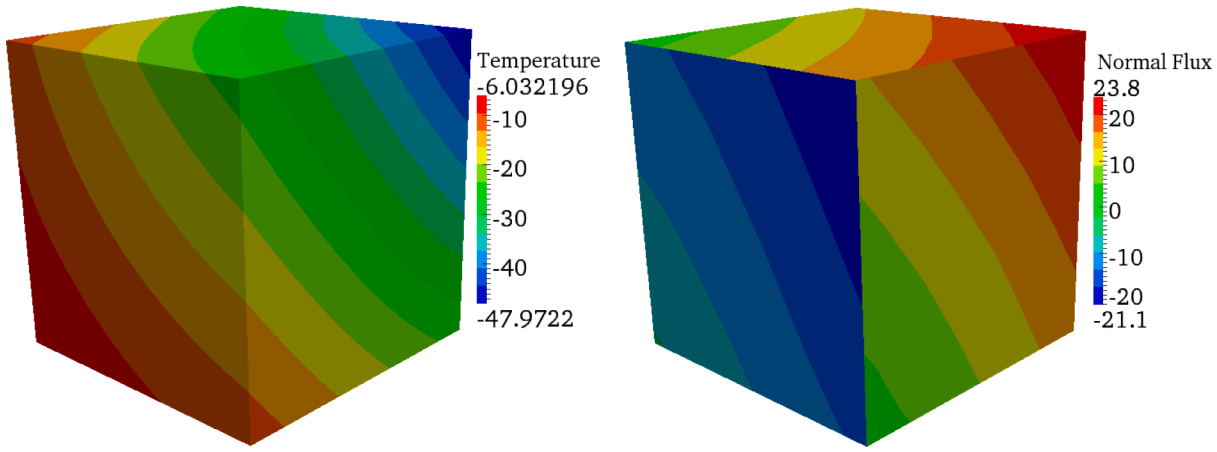

Fig. 16. Contours of (a) temperature and (b) normal flux of the inner cube.

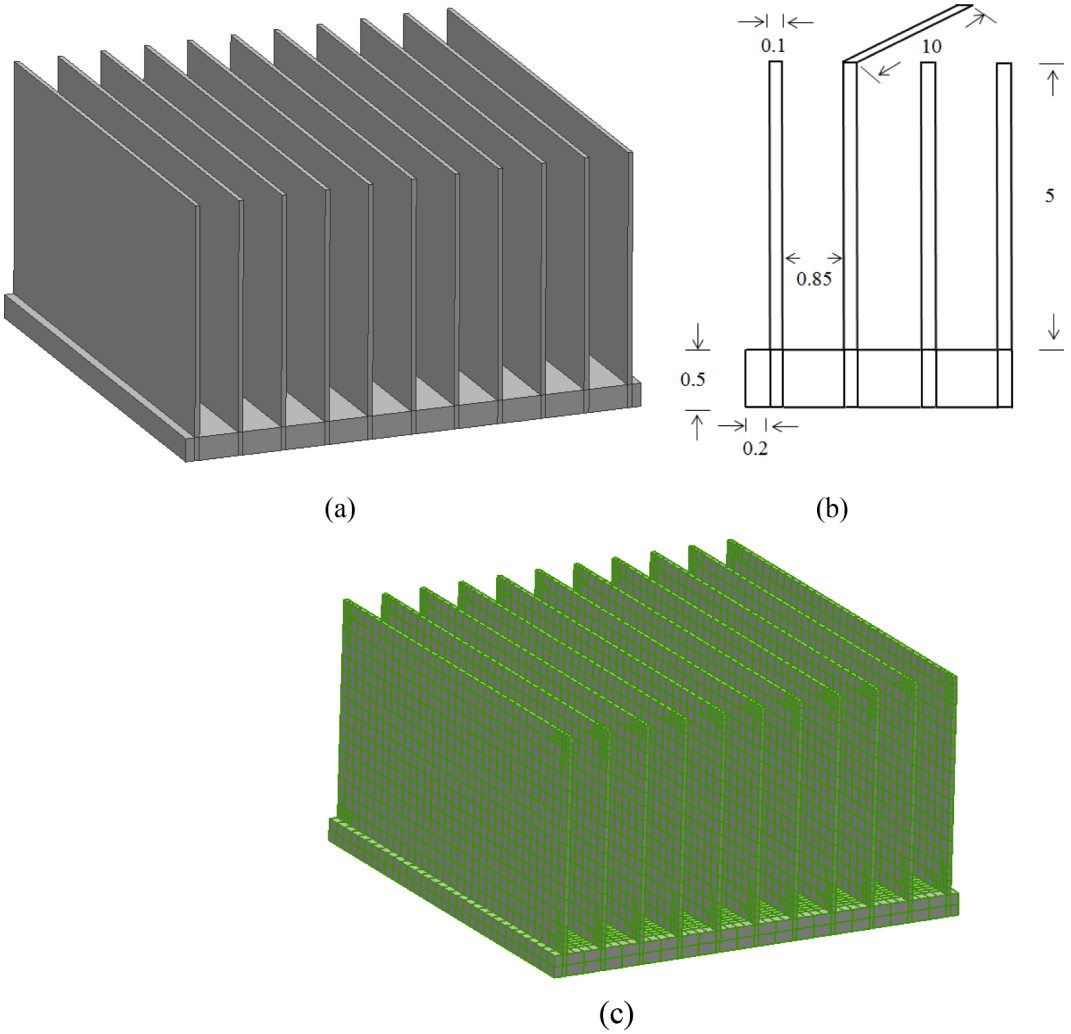

Fig. 17. (a) Sink with 34 subdomains and (b) the geometries and (c) binary tree meshes.

3.6. The choice of relaxation parameter

For different problems, the value of the relaxation parameter also varies. A reasonable relaxation parameter can ensure convergence or significantly reduce the number of iterations. To avoid prior determination of the relaxation parameter, a technique for dynamically selecting relaxation parameters is introduced. In the implementation, the corresponding error function is built according to the different DDM cases, and an expression for the relaxation parameter can be obtained by taking the extreme value for the error function. The value of the relaxation parameter in each iteration step can be calculated accordingly.

Taking the DDM using Neumann - Neumann interface conditions as

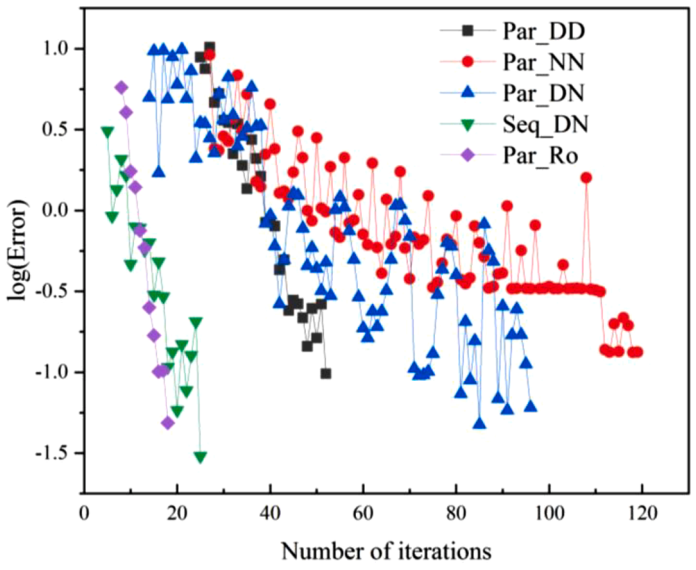

Fig. 18. The maximum interface error versus the number of iterations.

Table 4
Comparison of iterations and errors between the five DiBFM-DDMs.

Algorithm	nItrs	*Err_u*	*Err_q*
Par_DD	192	2.012E-5	1.704E-3
Par_NN	313	2.01E-5	1.68E-3
Par_DN	188	2.011E-5	1.771E-3
Seq_DN	55	2.061E-5	1.712E-3
Par_Ro	57	2.016E-5	1.674E-3

an example, the process is shown below:

Firstly, introducing the iterative equations:

$$\begin{aligned}\mathbf{q}_M^{nItr} &= \mathbf{q}_M^{nItr-1} + \beta\left(\mathbf{u}_S^{nItr-1} - \mathbf{u}_M^{nItr-1}\right)\\ \mathbf{q}_M^{nItr+1} &= \mathbf{q}_M^{nItr} + \beta\left(\mathbf{u}_S^{nItr} - \mathbf{u}_M^{nItr}\right)\end{aligned} \tag{20}$$

Then substituting Eq. (20) into the given error function as follows:

$$\begin{aligned}F(\beta) &= \| \mathbf{q}_M^{nItr+1} - \mathbf{q}_M^{nItr} \|_2^2\\ &= \| \mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1} + \beta\left[\left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right)\right] \|_2^2\\ &= \| \mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1} \|_2^2 + 2\beta\left(\mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1}\right)\left[\left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right)\right]\\ &\quad + \beta^2 \| \left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right) \|_2^2\end{aligned} \tag{21}$$

where $||||_2$ denote the l_2 norm.

Taking the partial derivative with respect to β yields:

$$\begin{aligned}F'(\beta) = 2\left(\mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1}\right)\left[\left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right)\right]\\ + 2\beta \| \left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right) \|_2^2\end{aligned} \tag{22}$$

Obviously,

$$F''(\beta) = 2 \| \left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right) \|_2^2 > 0 \tag{23}$$

Therefore setting $F'(\beta) = 0$, the expression for β can be obtained below

$$\begin{aligned}\beta = -\left(\mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1}\right)\left[\left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right)\right]\\ \Big/ \| \left(\mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}\right) - \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right) \|_2^2\end{aligned} \tag{24}$$

Similarly, the expressions for α, γ, ρ are given as follows, details can

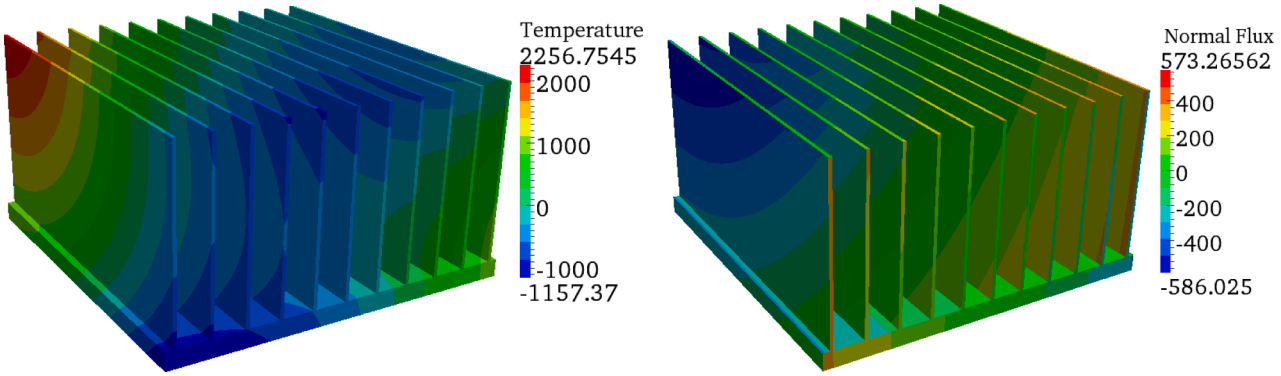

Fig. 19. Contours of (a) temperature and (b) normal flux of the sink.

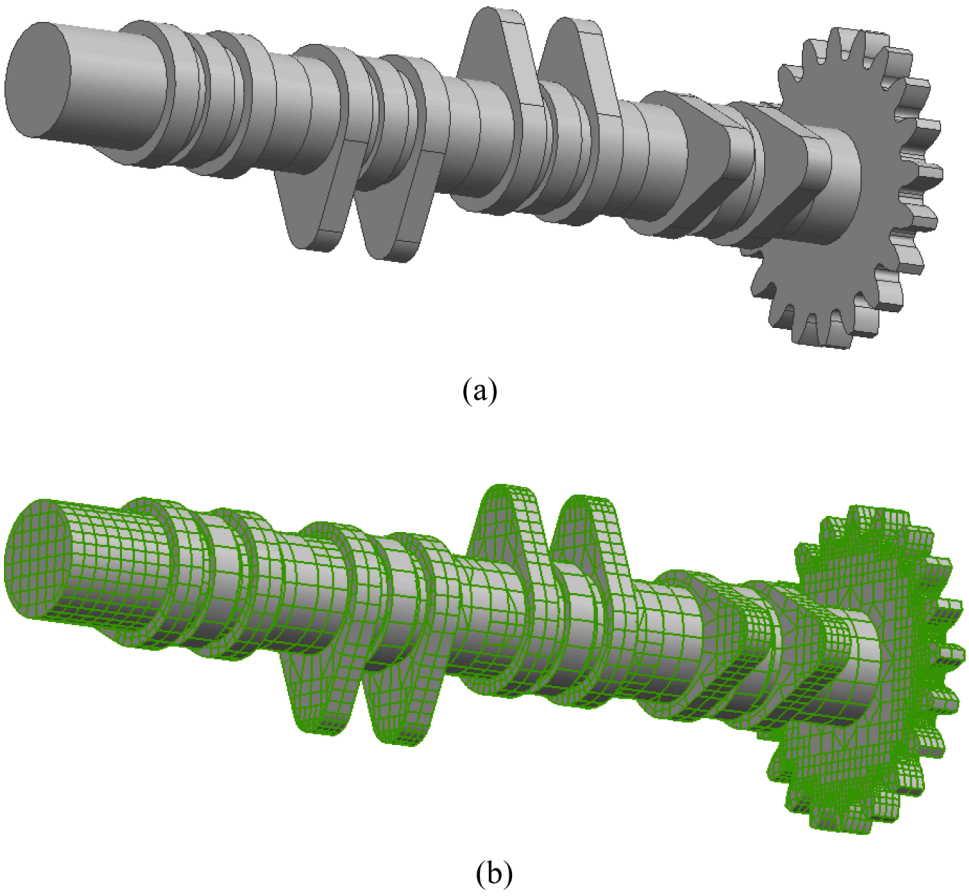

Fig. 20. (a) Camshaft with 5 subdomains and (b) its binary tree meshes.

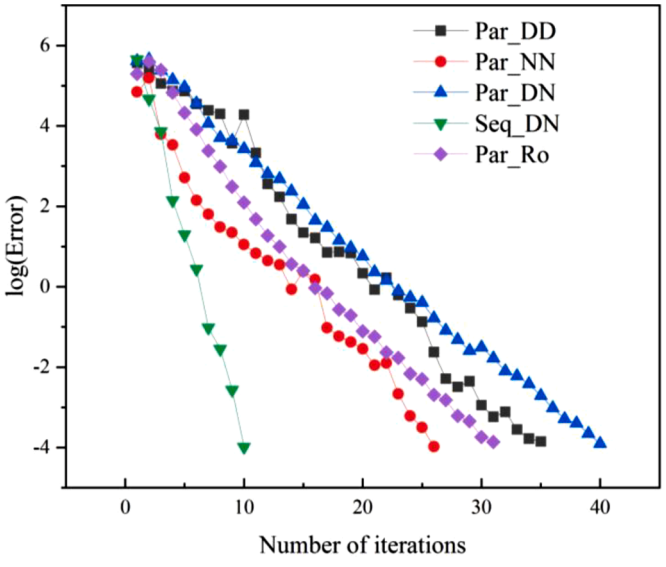

Fig. 21. The maximum interface error versus the number of iterations.

be referenced in the literature [23–24].

$$\alpha = \left(\mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}\right)\left[\left(\mathbf{q}_S^{nItr} - \mathbf{q}_S^{nItr-1}\right) + \left(\mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1}\right)\right] \Big/ \| \left(\mathbf{q}_S^{nItr} - \mathbf{q}_S^{nItr-1}\right) + \left(\mathbf{q}_M^{nItr} - \mathbf{q}_M^{nItr-1}\right) \|_2^2 \tag{25}$$

$$\gamma = < \mathbf{e}_M^{nItr}, \mathbf{e}_M^{nItr} - \mathbf{e}_S^{nItr} > \Big/ \| \mathbf{e}_M^{nItr} - \mathbf{e}_S^{nItr} \|_2^2 \tag{26}$$

Table 5
Comparison of iterations and errors between the five DiBFM-DDMs.

Algorithm	nItrs	*Err_u*	*Err_q*
Par_DD	35	3.650E-5	4.929E-4
Par_NN	26	3.109E-5	4.049E-4
Par_DN	40	2.920E-5	4.352E-4
Seq_DN	10	2.090E-5	3.305E-4
Par_Ro	31	2.209E-5	3.649E-4

$$\rho = \left[< \mathbf{e}_M^{nItr}, \mathbf{e}_M^{nItr} - \mathbf{e}_S^{nItr} > + < \mathbf{e}_M^{nItr-1}, \mathbf{e}_M^{nItr-1} - \mathbf{e}_S^{nItr-1} > \right] / \left(\| \mathbf{e}_M^{nItr} - \mathbf{e}_S^{nItr} \|_2^2 + | \mathbf{e}_M^{nItr-1} - \mathbf{e}_S^{nItr-1} \|_2^2 \right) \tag{27}$$

where $\mathbf{e}_M^{nItr} = \mathbf{u}_M^{nItr} - \mathbf{u}_M^{nItr-1}, \mathbf{e}_S^{nItr} = \mathbf{u}_S^{nItr} - \mathbf{u}_S^{nItr-1}$ and $< a, b >$ represents the inner product of a and b.

4. Numerical examples

To carry out the comparative study of the above DiBFM-DDMs, three numerical examples are given in this section. The first example considers the case of two subdomains. The effectiveness of handling interior cross-points problems is verified in the second example. The last example is presented to illustrate the capacity of DiBFM-DDMs to solve complex model. For the sake of simplification, the symbols 'Par_DD', 'Par_NN', 'Par_DN', 'Seq_DN' and 'Par_Ro' denote the parallel Dirichlet–Dirichlet, parallel Neumann–Neumann, parallel Dirichlet–Neumann, sequential Dirichlet–Neumann and parallel Robin–Robin scheme respectively. The convergence criterions are given in Section 3 and the tolerance $\varepsilon = 0.0001$ if not specified. The analytical boundary conditions of potential and flux are used for reference solutions and their forms are given below:

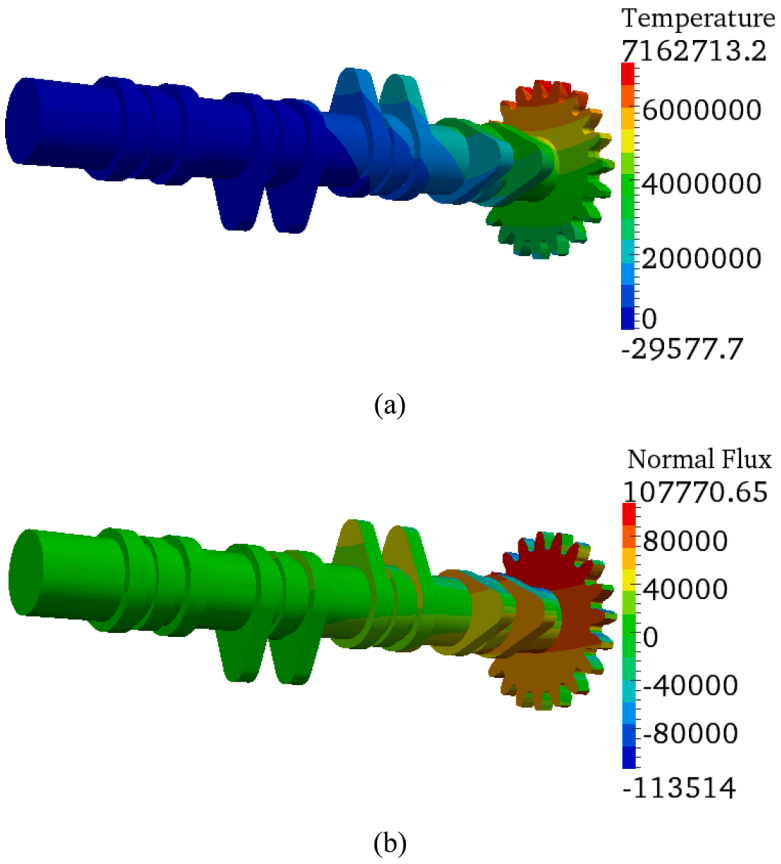

Fig. 22. Contours of (a) temperature and (b) normal flux of the camshaft.

$$u = x^3 + y^3 + z^3 - 3x^2y - 3y^2z - 3z^2x \tag{28}$$

$$q = -k\frac{\partial u}{\partial n} = -k\left(q_x n_x + q_y n_y + q_z n_z\right) \tag{29}$$

where

$$q_x = 3x^2 - 6xy - 3z^2, q_y = 3y^2 - 6yz - 3x^2, q_z = 3z^2 - 6xz - 3y^2 \tag{30}$$

and k stands for the heat conductivity in steady-state heat conduction problems.

The relative errors between the analytical and numerical solutions are evaluated by the following formula:

$$error = \frac{1}{|w^{(e)}|_{\max}}\sqrt{\frac{1}{M}\sum_{i=1}^{M}\left[w_i^{(e)} - w_i^{(c)}\right]^2} \tag{31}$$

where $|w^{(e)}|_{\max}$ represents the maximum node value, $\omega_i^{(e)}$ and $\omega_i^{(c)}$ are the analytical and numerical solutions, respectively. The relative errors of potential and flux are denoted by symbols *Err_u* and *Err_q* respectively.

4.1. Cuboid

Consider the multi-domain problem of cuboid firstly. The cuboid is decomposed into two cubes with side length of 1 m as shown in Fig. 7(a). The binary tree meshes are applied to discrete the boundaries as shown in Fig. 7(b) and are similar to the meshes obtained by mapping method. Using the analytical boundary conditions defined in Eqs. (28,29), the Dirichlet boundary conditions are imposed to the left and right sides of the cuboid and the Neumann boundary conditions are applied to remaining surfaces. The total number of elements, source and virtual nodes is 1200, 2192 and 5436, respectively.

The fixed relaxation parameter is investigated firstly, Fig. 8 gives the number of iterations under different relaxation parameters. The results show that, for each algorithm the relaxation parameters have applicable range to achieve convergence, a reasonable value can yield a small number of iterations. Obviously, the relaxation parameters range of the Par_DD is much narrower than that of the other algorithms. Fig. 9 shows the number of iterations under different coupling coefficients in Par_Ro, and the coupling coefficients $\lambda = 0.5$ is used if not specified in the following calculation.

By introducing the dynamic relaxation parameter, there is no prior determination is required in the iteration process. To illustrate the validity and superiority of dynamic relaxation parameter, the comparison of iterations between optimal and dynamic relaxation parameter is presented in Table. 2. Fig. 10 shows the distribution of temperature and flux of the cuboid. 'nItrs' denote the number of iterations

Examining Table 2 reveals that the convergence rate of dynamic relaxation parameter approaches or even exceeds that of the optimal one selected from the fixed relaxation parameters, and all algorithms are close in terms of accuracy. The Seq_DN converges the fastest and the Par_DD is the slowest. The dynamic relaxation parameters are used by default unless otherwise specified in the following results.

Next, the effect of mesh size on the convergence is considered and depicted in Fig. 11. It can be seen that the Par_DD is sensitive to the mesh size and its number of iterations is roughly positively correlated with the mesh size, while the mesh size has little effect on the convergence of the other algorithms.

The number of iterations varies with different tolerances as shown in Fig. 12. The change in the slope of the line implies a change in convergence rate, the Par_DN and Seq_DN can maintain a stable convergence rate while the other algorithms have fluctuations in the convergence rate.

Fig. 13 gives the effect of initial guess on the convergence. The close

the initial guess is to the theoretical solution, the faster the convergence achieve, and the Seq_DN is more robust than the others.

4.2. Cube

The cube consisting of 27 subdomains are investigated in this example, each subdomain is a small cube with side length of 1 m as shown in Fig. 14(a) and its discretization with binary tree meshes are depicted in Fig. 14(b). It should be noted that there is an inner cube whose boundaries are all inside, the Par_NN is no longer applicable to this model since the Neumann boundary conditions will be imposed on all interfaces of the inner cube, and the coefficient matrix is singular. In this case, the Dirichlet boundary condition defined in Eq. (28) is applied to all the surfaces of the cube, the dynamic relaxation parameters are used in iteration process. The total number of elements, source and virtual nodes is 4050, 8658 and 21,888, respectively.

Fig. 15 shows the maximum interface error versus the number of iterations, indicating the convergence process is oscillating, the steeper the curve, the faster the error drops, and the faster the convergence, the Par_Ro converges the fastest and the Par_DD is the slowest. Table. 3 gives the relative errors of temperature and flux of the inner cube and the overall rate of convergence. Solutions obtained by all algorithms agree well with analytical solutions. The distribution of temperature and flux of the inner cube is depicted in Fig. 16.

Compared to the previous example, the number of interfaces has increased, which also leads to an increase in the required number of iterations, and the interior cross-points exist on common vertices of inner subdomain, this example also demonstrates there is no need for a specialized preprocessor to handle the problem of interior cross-points in DiBFM-DDMs.

4.3. Sink

The sink decomposed into 34 subdomains is studied in this example. The CAD model of the sink and its geometry, as well as its binary tree mesh are depicted in Fig. 17. There are 33 interfaces and the Dirichlet boundary condition defined in Eq. (28) is applied to all the surfaces of the sink. The total number of elements, source and virtual nodes is 26,212, 45,352 and 114,288, respectively.

Fig. 18 shows the maximum interface error versus the number of iterations, the Seq_DN and Par_Ro still converge significantly faster than the others. It can be seen from Table 4 that results of all algorithms are basically consistent with the analytical solutions and their differences are not significant. The distribution of temperature and flux of the sink is depicted in Fig. 19.

4.4. Camshaft

The camshaft is decomposed into 5 subdomains in this example and there are 4 interfaces. The CAD model of the camshaft and its binary tree mesh are shown in Fig. 20. The Dirichlet boundary condition defined in Eq. (28) is applied to all the surfaces of the sink. The total number of elements, source and virtual nodes is 9204, 12,749 and 49,612, respectively.

Fig. 21 shows the maximum interface error versus the number of iterations, the Seq_DN converge faster than the others. It can be seen from Table 5 that results of all algorithms are basically consistent with the analytical solutions. The distribution of temperature and flux of the Camshaft is depicted in Fig. 22.

5. Discussion and conclusions

Five non-overlapping domain decomposition algorithms in the frame of dual interpolation boundary face method are developed and investigated for solving multi-domain potential problems. The main differences between these algorithms are the transmission conditions on the interfaces between subdomains. For most DDMs, a reasonable relaxation parameter can ensure or accelerate the convergence. Instead of predetermining a constant relaxation parameter, the dynamic relaxation parameters are derived based on error analysis in each iterative step to achieve an efficient convergence. Besides, by coupling DDMs with the DiBFM which boundary integral equation is collocated on nodes inside elements, the problem of interior cross-points is naturally avoided. Numerical examples illustrate that the sequential Dirichlet- Neumann and parallel Robin- Robin methods, in general, have better performance than the others, and the parallel Dirichlet- Dirichlet method is relatively poor. This study will be extended to multi-domain elasticity problems and contact problems in the next step of work.

CRediT authorship contribution statement

Rongxiong Xiao: Writing – original draft, Investigation, Formal analysis, Data curation, Conceptualization. **Jianming Zhang:** Project administration, Methodology, Formal analysis, Conceptualization. **Yang Yang:** Writing – review & editing, Validation, Supervision. **Chong Zhang:** Writing – review & editing, Visualization, Supervision.

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. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by National Natural Science Foundation of China under grant numbers 12172126, 11972010 and 12272160.

References

[1] Chai PF, Zhang JM, Xiao RX, et al. A multi-domain BEM based on dual interpolation boundary face method for 3D potential problem [J]. Acta Mech 2022: 1–19.
[2] Dong CY. A interface integral formulation of heat energy calculation of steady state heat conduction in hererogeneous media [J]. Int J Heat Mass Transf 2015;90: 314–22.
[3] Liu YJ. A fast multipole boundary element method for 2D multi-domain elastostatic problems based on a dual BIE formulation [J]. Comput Mech 2008;42:761–73.
[4] Chai PF, Zhang JM, Xiao RX, et al. A multi-domain BEM based on dual interpolation boundary face method for 3D elasticity problem [J]. Eng Anal Boundary Elements 2022;143:568–78.
[5] Gao XW, Guo L, Zhang C. Three-step multi-domain BEM solver for nonhomogeneous material problems[J]. Eng Anal Bound Elem 2007;31:965–73.
[6] Kane JH, Kumar BLK, Saigal S. An arbitrary condensing, noncondensing solution strategy for large scale, multi-zone boundary element analysis [J]. Comput Methods Appl Mech Eng 1990;79:219–44.
[7] Schwarz HA. Gesammelte mathematische abhandlungen [J]. Vierteljahrsschrift der naturforschenden gesselschaft in Zurich 1870;15:272–86.
[8] Smith B, Bjorstad P, Gropp W. Domain decomposition [M]. Cambridge University Press; 1996.
[9] Lions PL. On the Schwarz alternating methods III: a variant for nonoverlapping subdomains. In: Third International Symposium on Decomposition Methods for Partial Differential Equations. SIAM, Philadelohia; 1990. p. 202–23.
[10] Guo W, Hou LS. Generalizations and accelerations of Lions'nonoverlapping domain decomposition for linear elliptic PDE. SIAM 2003;41:2056–80.
[11] Dolean V, Jolivet P, Nataf F. An introduction to domain decomposition methods: algorithms, theory and parallel implementation. In: ; 2015.

[12] Elleithy WM, Tanaka M. Interface relaxation algorithms for coupling the FEM and BEM [J]. Comput Methods Appl Mech Eng 2003;192:2977–92.
[13] Ingber MS, Tanski JA, Alsing P. A domain decomposition tool for boundary element methods [J]. Eng Anal Bound Elem 2007;31:890–6.
[14] Francois S, Coulier P, Degrande G. Finite element-boundary coupling algorithms for transient elastodynamics [J]. Eng Anal Bound Elem 2015;55:104–21.
[15] Bakalakos S, Georgioudakis M, Papadrakakis M. Domain decomposition methods for 3D crack propagation problems using XFEM [J]. Comput Methods Appl Mech Eng 2022;402:115390.
[16] Peng Z, Wang X, Lee JF. Integral equation based domain decomposition method for solving electromagnetic wave scattering from non-penetrable objects [J]. IEEE Trans, Antennas Propag 2011;59:3328–38.
[17] Sikora J, Grzywacz T. Domain decomposition method for diffuse optical tomography problems [J]. Eng Anal Bound Elem 2012;36:1005–13.
[18] Erhart K, Divo E, Kassab Alain J. A parallel domain decomposition boundary element method approach for the solution of large-scale transient heat conduction problems [J]. Eng Anal Bound Elem 2006;30:553–63.
[19] Zhang JM, He R, Lin WC, et al. A dual interpolation boundary face method with Hermite-type approximation for elasticity problems [J]. Eur J Mechan /A Solids 2020;82:104005.
[20] Zhang JM, Xiao RX, Wen PH, et al. Dual interpolation boundary face method for 3-D potential problem based on binary tree girds [J]. Comput Methods Appl Mech Eng 2022;390:114432.
[21] Zhang JM, Qin XY, Han X. A boundary face method for potential problems in three dimensions [J]. Int J Numer Methods Eng 2009;80:320–37.
[22] Yu B, Gao G, Gong Y. IG-DRBEM of three-dimensional transient heat conduction problems [J]. Eng Anal Bound Elem 2021;128:298–309.
[23] Lin CC, Lawton EC, Caliendo JA, et al. An iterative finite element-boundary element algorithm [J]. Comput Struct 1996;59:899–909.
[24] Li MJ. The coupling method and domain decomposition method based on boundary element method and meshless local Petrov-Galerkin method [D]. Chongqing University; 2009.