Comput Mech (2016) 58:533–548
DOI 10.1007/s00466-016-1304-0

ORIGINAL PAPER

A new simple multidomain fast multipole boundary element method

S. Huang[1] · **Y. J. Liu**[1,2]

Received: 10 December 2015 / Accepted: 12 May 2016 / Published online: 9 June 2016

Abstract A simple multidomain fast multipole boundary element method (BEM) for solving potential problems is presented in this paper, which can be applied to solve a true multidomain problem or a large-scale single domain problem using the domain decomposition technique. In this multidomain BEM, the coefficient matrix is formed simply by assembling the coefficient matrices of each subdomain and the interface conditions between subdomains without eliminating any unknown variables on the interfaces. Compared with other conventional multidomain BEM approaches, this new approach is more efficient with the fast multipole method, regardless how the subdomains are connected. Instead of solving the linear system of equations directly, the entire coefficient matrix is partitioned and decomposed using Schur complement in this new approach. Numerical results show that the new multidomain fast multipole BEM uses fewer iterations in most cases with the iterative equation solver and less CPU time than the traditional fast multipole BEM in solving large-scale BEM models. A large-scale fuel cell model with more than 6 million elements was solved successfully on a cluster within 3 h using the new multidomain fast multipole BEM.

Keywords Multidomain problems · Domain decomposition · Boundary element method · Fast multipole method

✉ Y. J. Liu
Yijun.Liu@uc.edu

[1] Mechanical Engineering, University of Cincinnati, P. O. Box 210072, Cincinnati, OH 45221-0072, USA

[2] Institute for Computational Mechanics and Its Applications, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China

1 Introduction

The multidomain boundary element method (BEM) has been studied since 1980s. A comprehensive introduction of the multidomain BEM can be found in Ref. [1]. Many researchers have proposed different multidomain BEM approaches for modeling potential problems [2–7], elastostatic problems [8–15], acoustic problems [16–20], elastodynamic problems [21–23], fluid mechanics [24–27] and fracture mechanics problems [28–30]. With these approaches, the entire domain is divided into some subdomains and the boundary integral equation (BIE) are applied to each domain. The total number of unknown degrees of freedom (DOFs) increases because there are some new elements on the interface. Most approaches use the interface conditions to eliminate the additional DOFs on the interface [10,31], which may require extra effort in rearrangement of the coefficient matrices and make it difficult to apply the fast algorithms and parallel computing for the general cases. Recently, an interface integral BEM for multidomain problems is derived in [2,11]. This method combines the integral equations of all subdomains and the interfaces. Therefore, only one boundary integral equation is needed.

Multidomain BEM approaches developed for problems with different material properties in different subdomains can also be used to solve problems in domains with a single material. This approach can be regarded as a domain decomposition technique for solving large-scale single domain problems. Rigby and Aliabadi [32] used a multidomain BEM to solve the problems with a uniform material property in the entire domain and found out that it was faster than the traditional single domain BEM approach for the cases they studied. The idea behind this approach is to solve large-scale single domain problems using the multidomain BEM approach that can facilitate easy implementation of paral-

lel computing. Hsiao and Wendland also did some early exploration [33,34] on the domain decomposition with the BIE. Recently, Langer and Steinbach [35–37] introduced the boundary element tearing and interconnecting (BETI) method and applied the fast multipole method (FMM) [38–43] to accelerate the solution. Pechstein [44] extended the BETI method to problems with unbounded domains. Some researchers also proposed other approaches of the domain decomposition with the BEM. Ingber, Tanski and Alsing [45] proposed a domain decomposition tool for the BEM based on a graph partitioning algorithm. Bui and Popov [46] developed a numerical approach based on the domain decomposition boundary element method with overlapping subdomains.

In this paper, we present a new and general formulation of the multidomain fast multipole BEM. In this method, the BEM coefficient matrix is formed simply by assembling the local coefficient matrices of each subdomain and the interface conditions between attached subdomains without eliminating the unknown variables on the interface. Instead of solving the large linear system of equations directly, the entire coefficient matrix is partitioned and decomposed using Schur complement in this new approach. Compared to most of the multidomain BEM approaches in the literature, this new simple approach is shown to be more efficient with the fast multipole method and suitable for parallel computing, regardless how the subdomains are attached to each other.

The paper is organized as follows: In Sect. 2, the BEM formulation for potential problems are reviewed. In Sect. 3, the new formulation of the multidomain fast multipole BEM is proposed. The overall algorithm is shown in Sect. 4. Some numerical examples are given in Sect. 5, and some discussions are provided in Sect. 6.

2 BIE formulation for potential problems

Consider a potential problem in a 2D or 3D domain V with the boundary S . The governing equation of the potential problem is:

$$\nabla^2 \phi(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in V, \tag{1}$$

and the boundary conditions are:

$$\begin{aligned} \phi(\mathbf{x}) &= \bar{\phi}(\mathbf{x}), \quad \forall \mathbf{x} \in S_\phi, \\ q(\mathbf{x}) &= \bar{q}(\mathbf{x}), \quad \forall \mathbf{x} \in S_q, \end{aligned} \tag{2}$$

where $\phi(\mathbf{x})$ is the potential and $q(\mathbf{x})$ is the normal derivative of $\phi(\mathbf{x})$; $\bar{\phi}(\mathbf{x})$ and $\bar{q}(\mathbf{x})$ are two given functions on S_ϕ and S_q, respectively; $S = S_\phi \cup S_q$ and $S_\phi \cap S_q = \emptyset$.

With the help of Green's second identity, the corresponding conventional boundary integral equation (CBIE) for potential problems is [43]:

$$\begin{aligned} \frac{1}{2}\phi(\mathbf{x}) = &\int_S G(\mathbf{x},\mathbf{y}) q(\mathbf{y}) dS \\ &- \int_S F(\mathbf{x},\mathbf{y}) \phi(\mathbf{y}) dS, \quad \mathbf{x} \in S, \end{aligned} \tag{3}$$

where $G(\mathbf{x},\mathbf{y})$ is the fundamental solution, with $\mathbf{x}$ being the source (collocation) point and $\mathbf{y}$ the field (integration) point. For 3D potential problems, we have:

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi r}, \quad r = |\mathbf{x}-\mathbf{y}|. \tag{4}$$

$F(\mathbf{x},\mathbf{y})$ is the normal derivative of $G(\mathbf{x},\mathbf{y})$ at the field (integration) point $\mathbf{y}$.

Taking the normal derivative of Eq. (3) at $\mathbf{x}$ gives the hypersingular boundary integral equation (HBIE):

$$\begin{aligned} \frac{1}{2}q(\mathbf{x}) = &\int_S K(\mathbf{x},\mathbf{y}) q(\mathbf{y}) dS \\ &- \int_S H(\mathbf{x},\mathbf{y}) \phi(\mathbf{y}) dS, \quad \mathbf{x} \in S, \end{aligned} \tag{5}$$

where

$$K(\mathbf{x},\mathbf{y}) = \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{x})}, \quad H(\mathbf{x},\mathbf{y}) = \frac{\partial F(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{x})}.$$

It has been demonstrated that (see, e.g., Ref. [43]) a linear combination of CBIE and HBIE (dual BIE formulation) provides better conditioning for solving regular domain problems as well as crack or thin-shape domain problems. In this paper, only the CBIE formulation is employed.

3 New multidomain BEM formulation

For the conventional BEM for single-domain problems, the entire boundary S is discretized with some boundary elements. The next step is to calculate the coefficient matrix and right-hand-side vector. The last step is to solve it by using a direct solver. The conventional BEM works well when the size of the problem is not large (for example, with the number of DOFs less than ten thousands). Fast algorithms such as the fast multipole method [38–43] and adaptive cross approximation (ACA) method [47] can speed up the BEM in solving large-scale problems (for example, with a few million DOFs on a PC) without constructing the coefficient matrix explicitly. However, the BEM is still difficult to be applied to solve the problems with a larger number of DOFs (such as 100 million DOFs). The reason is that for large-scale problems, the conditioning of the coefficient matrix often deteriorates, which can lead to large numbers of iterations using iterative solvers and propagation of the errors.

A natural idea of solving larger scale problems is to use the domain decomposition concept or the multidomain BEM.

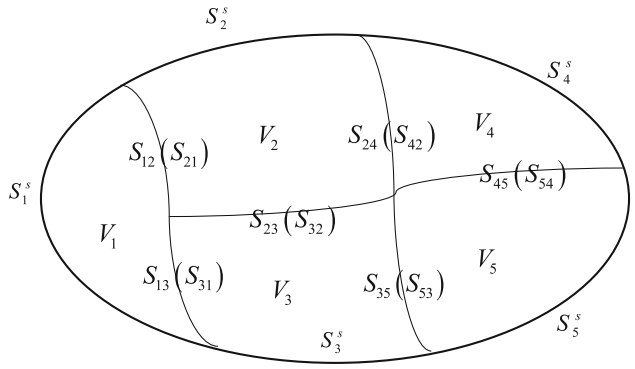

Fig. 1 Finite domain decomposition

In the following we first describe a new multidomain BEM formulation and then discuss its implementation.

First, we divide the entire problem domain V into M non-overlapping subdomains V_i as (Fig. 1):

$$V = \bigcup_{i=1}^{M} V_i, \quad S_i = \partial V_i.$$

where S_i is the boundary of the subdomain V_i; S_{ij} is the interface of V_i and V_j. Let S_i^s be the boundary of subdomain V_i excluding the interface and S_i^I as the group of all the interfaces of subdomain V_i. For subdomain V_i, the local field ϕ_i satisfies the governing equation (1) in domain V_i. The local boundary conditions for ϕ_i are:

$$\begin{aligned} \phi_i(\mathbf{x}) &= \overline{\phi_i}(\mathbf{x}), \quad \forall \mathbf{x} \in S_i^{\phi} = S_i \cap S_{\phi}, \\ q_i(\mathbf{x}) &= \overline{q_i}(\mathbf{x}), \quad \forall \mathbf{x} \in S_i^{q} = S_i \cap S_q, \end{aligned} \tag{6}$$

for $i = 1, 2, \ldots, M$, where q_i is the normal derivative of ϕ_i. Additional interface conditions are required to relate the new unknown variables on the interfaces. For potential problems, the interface conditions on interface S_{ij} are:

$$\phi_i(\mathbf{x}) = \phi_j(\mathbf{x}), \quad q_i(\mathbf{x}) = -q_j(\mathbf{x}), \quad \forall \mathbf{x} \in S_{ij}. \tag{7}$$

Note that the interface conditions are not unique. That is, any linearly independent combinations of the two expressions in Eq. (7) can be used as the interface conditions. Different forms of the interface conditions can influence the conditioning of the final coefficient matrix. In this work, the following form of the interface conditions is applied:

$$\begin{aligned} &\phi_i(\mathbf{x}) - \phi_j(\mathbf{x}) - q_i(\mathbf{x}) - q_j(\mathbf{x}) = 0, \\ &-\phi_i(\mathbf{x}) + \phi_j(\mathbf{x}) - q_i(\mathbf{x}) - q_j(\mathbf{x}) = 0, \quad \mathbf{x} \in S_{ij}. \end{aligned} \tag{8}$$

Now the original single-domain problem is divided into a group of local problems in M subdomains. Applying the BIE (Eqs. (3) or (5), or a linear combination of the two) for each subdomain, the final coefficient matrix of the BEM equations can be arranged without eliminating the duplicated DOFs on the interfaces as follows:

$$\begin{array}{l} \text{Subdomain } V_1 \\ \\ \text{Subdomain } V_2 \\ \\ \text{Subdomain } V_3 \\ \\ \vdots \\ \text{Subdomain } V_M \\ \\ \end{array} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_1^I & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{1,1} & \mathbf{0} & \mathbf{I}_{1,2} & \mathbf{0} & \mathbf{I}_{1,3} & \cdots & \mathbf{0} & \mathbf{I}_{1,M} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \mathbf{A}_2^I & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{2,1} & \mathbf{0} & \mathbf{I}_{2,2} & \mathbf{0} & \mathbf{I}_{2,3} & \cdots & \mathbf{0} & \mathbf{I}_{2,M} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & \mathbf{A}_3^I & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3,1} & \mathbf{0} & \mathbf{I}_{3,2} & \mathbf{0} & \mathbf{I}_{3,3} & \cdots & \mathbf{0} & \mathbf{I}_{3,M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_M & \mathbf{A}_M^I \\ \mathbf{0} & \mathbf{I}_{M,1} & \mathbf{0} & \mathbf{I}_{M,2} & \mathbf{0} & \mathbf{I}_{M,3} & \cdots & \mathbf{0} & \mathbf{I}_{M,M} \end{bmatrix}$$

$$\begin{Bmatrix} \boldsymbol{\lambda}_1^S \\ \boldsymbol{\lambda}_1^I \\ \boldsymbol{\lambda}_2^S \\ \boldsymbol{\lambda}_2^I \\ \boldsymbol{\lambda}_3^S \\ \boldsymbol{\lambda}_3^I \\ \vdots \\ \boldsymbol{\lambda}_M^S \\ \boldsymbol{\lambda}_M^I \end{Bmatrix} = \begin{Bmatrix} \mathbf{b}_1 \\ \mathbf{0} \\ \mathbf{b}_2 \\ \mathbf{0} \\ \mathbf{b}_3 \\ \mathbf{0} \\ \vdots \\ \mathbf{b}_M \\ \mathbf{0} \end{Bmatrix} \tag{9}$$

where $\boldsymbol{\lambda}_i^S$ and $\boldsymbol{\lambda}_i^I$ are vectors of unknowns on S_i^s and S_i^I for subdomain V_i, respectively. Because both potential and its normal derivative are unknown on the interfaces, $\boldsymbol{\lambda}_i^I$ can be further divided as two components: $\boldsymbol{\lambda}_{i,\phi}^I$ and $\boldsymbol{\lambda}_{i,q}^I$, where $\boldsymbol{\lambda}_{i,\phi}^I$ is the unknown potential on the interfaces and $\boldsymbol{\lambda}_{i,q}^I$ is the unknown normal derivatives of potential on the interfaces. $\mathbf{A}_i$ and $\mathbf{A}_i^I$ are the coefficient matrices corresponding to the unknown vectors $\boldsymbol{\lambda}_i^S$ and $\boldsymbol{\lambda}_i^I$, respectively. $\mathbf{b}_i$ is the right-hand-side vector. $\mathbf{I}_{i,j}$ are the matrices associated with the interface conditions. It is worth to mention that $\mathbf{I}_{i,j}$ usually are not the identity matrices. The entries of $\mathbf{I}_{i,j}$ consist of 0, 1 or -1. The exact form of $\mathbf{I}_{i,j}$ depends on the choice and implementation of interface conditions. In this paper, the first and second equations of Eq. (8) are arranged under the subdomain V_i and V_j, respectively, in Eq. (9). Using fast multipole method to solve Eq. (9) iteratively may need a good preconditioner to achieve high efficiency. In [48], Zhang et al. proposed a re-numbering strategy to improve the conditioning of the coefficient matrix for multidomain problems. In this paper, a new approach of matrix decomposition is proposed

to improve the efficiency of solving multidomain problems.

Based on the locations of the source point and field point, $\mathbf{A}_i$ and $\mathbf{A}_i^I$ can be partitioned in the following form:

$$\begin{bmatrix} \mathbf{A}_i & \mathbf{A}_i^I \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i^{SS} & \mathbf{A}_i^{SI} \\ \mathbf{A}_i^{IS} & \mathbf{A}_i^{II} \end{bmatrix} \tag{10}$$

where $\mathbf{A}_i^{SS}$ is obtained by calculating the integral when both the source point $\mathbf{x}$ and field point $\mathbf{y}$ are on S_i^S; $\mathbf{A}_i^{SI}$ is obtained by calculating the integral when the source point $\mathbf{x}$ is on S_i^S and field point $\mathbf{y}$ is on S_i^I; $\mathbf{A}_i^{IS}$ is obtained by calculating the integral when the source point $\mathbf{x}$ is on S_i^I and field point $\mathbf{y}$ is on S_i^S; $\mathbf{A}_i^{II}$ is obtained by calculating the integral when both the source point $\mathbf{x}$ and field point $\mathbf{y}$ are on S_i^I. Set the number of elements on S_i^S and S_i^I as n_i^S and n_i^I, respectively. $n_i = n_i^S + n_i^I$. The sizes of $\mathbf{A}_i^{SS}$, $\mathbf{A}_i^{SI}$, $\mathbf{A}_i^{IS}$ and $\mathbf{A}_i^{II}$ are $n_i^S \times n_i^S$, $n_i^S \times 2n_i^I$, $n_i^I \times n_i^S$ and $n_i^I \times 2n_i^I$, respectively. $\mathbf{A}_i^{SI}$ and $\mathbf{A}_i^{II}$ are further divided into two submatrices as follows:

$$\mathbf{A}_i^{SI} = \begin{bmatrix} \mathbf{A}_{i,\phi}^{SI} & \mathbf{A}_{i,q}^{SI} \end{bmatrix}, \quad \mathbf{A}_i^{II} = \begin{bmatrix} \mathbf{A}_{i,\phi}^{II} & \mathbf{A}_{i,q}^{II} \end{bmatrix},$$

where $\mathbf{A}_{i,\phi}^{SI}$ and $\mathbf{A}_{i,q}^{SI}$ are $n_i^S \times n_i^I$ matrices; $\mathbf{A}_{i,\phi}^{II}$ and $\mathbf{A}_{i,q}^{II}$ are $n_i^I \times n_i^I$ matrices; $\mathbf{A}_{i,\phi}^{SI}$ and $\mathbf{A}_{i,\phi}^{II}$ are the matrices corresponding to the unknown potential on the interfaces of subdomain V_i. $\mathbf{A}_{i,q}^{SI}$ and $\mathbf{A}_{i,q}^{II}$ are corresponding to the unknown normal derivative of potential on the interfaces of subdomain V_i. Define

$$\mathbf{A'}_i = \begin{bmatrix} \mathbf{A}_i^{SS} & \mathbf{A}_{i,\phi}^{SI} \\ \mathbf{A}_i^{IS} & \mathbf{A}_{i,\phi}^{II} \end{bmatrix}, \quad \mathbf{A'}_i^I = \begin{bmatrix} \mathbf{A}_{i,q}^{SI} \\ \mathbf{A}_{i,q}^{II} \end{bmatrix} \tag{11}$$

As it is stated before, the first and second equations of Eq. (8) are arranged under the subdomain V_i and V_j, respectively, in Eq. (9). Therefore, $\mathbf{I}_{i,j}$ also can be divided into two submatrices as: $\mathbf{I}_{i,j} = \begin{bmatrix} (-1)^{\delta_{ij}+1}\mathbf{I}_{i,j}^{\phi} & -\mathbf{I}_{i,j}^{q} \end{bmatrix}$ where δ_{ij} is the Kronecker delta function. The coefficients in front of unknown potential and normal derivative of potential on the interfaces are arranged in $\mathbf{I}_{i,j}^{\phi}$ and $\mathbf{I}_{i,j}^{q}$, respectively. If subdomain V_i and V_j do not have interfaces, all entries of $\mathbf{I}_{i,j}^{\phi}$ and $\mathbf{I}_{i,j}^{q}$ are zero. Otherwise, some entries of $\mathbf{I}_{i,j}^{\phi}$ and $\mathbf{I}_{i,j}^{q}$ are 1. There is at most one non-zero entry in each row of $\mathbf{I}_{i,j}^{\phi}$ and $\mathbf{I}_{i,j}^{q}$. It is worth to mention that $\mathbf{I}_{i,j}^{\phi}$ and $\mathbf{I}_{i,j}^{q}$ also usually are not identity matrices.

We re-arrange Eq. (9) as follows:

$$\begin{bmatrix}
\mathbf{A'}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A'}_1^I & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{A'}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{A'}_2^I & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A'}_3 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A'}_3^I & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_M & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A'}_M^I \\
\mathbf{I}_{1,1}^{\phi} & -\mathbf{I}_{1,2}^{\phi} & -\mathbf{I}_{1,3}^{\phi} & \cdots & -\mathbf{I}_{1,M}^{\phi} & -\mathbf{I}_{1,1}^{q} & -\mathbf{I}_{1,2}^{q} & -\mathbf{I}_{1,3}^{q} & \cdots & -\mathbf{I}_{1,M}^{q} \\
-\mathbf{I}_{2,1}^{\phi} & \mathbf{I}_{2,2}^{\phi} & -\mathbf{I}_{2,3}^{\phi} & \cdots & -\mathbf{I}_{2,M}^{\phi} & -\mathbf{I}_{2,1}^{q} & -\mathbf{I}_{2,2}^{q} & -\mathbf{I}_{2,3}^{q} & \cdots & -\mathbf{I}_{2,M}^{q} \\
-\mathbf{I}_{3,1}^{\phi} & -\mathbf{I}_{3,2}^{\phi} & \mathbf{I}_{3,3}^{\phi} & \cdots & -\mathbf{I}_{3,M}^{\phi} & -\mathbf{I}_{3,1}^{q} & -\mathbf{I}_{3,2}^{q} & -\mathbf{I}_{3,3}^{q} & \cdots & -\mathbf{I}_{3,M}^{q} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
-\mathbf{I}_{M,1}^{\phi} & -\mathbf{I}_{M,2}^{\phi} & -\mathbf{I}_{M,3}^{\phi} & \cdots & -\mathbf{I}_{M,M}^{\phi} & -\mathbf{I}_{M,1}^{q} & -\mathbf{I}_{M,2}^{q} & -\mathbf{I}_{M,3}^{q} & \cdots & -\mathbf{I}_{M,M}^{q}
\end{bmatrix}
\begin{Bmatrix}
\boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \\ \vdots \\ \boldsymbol{\lambda}_M \\ \boldsymbol{\lambda}_{1,q}^I \\ \boldsymbol{\lambda}_{2,q}^I \\ \boldsymbol{\lambda}_{3,q}^I \\ \vdots \\ \boldsymbol{\lambda}_{M,q}^I
\end{Bmatrix} =
\begin{Bmatrix}
\mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_M \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0}
\end{Bmatrix} \tag{12}$$

where $\boldsymbol{\lambda}_i = \begin{Bmatrix} \boldsymbol{\lambda}_i^S \\ \boldsymbol{\lambda}_{i,\phi}^I \end{Bmatrix}$; $\mathbf{I}_{i,j}^{\phi}$ and $\mathbf{I}_{i,j}^{q}$ are the matrices representing the interface conditions of subdomains V_i and V_j.

Define

$$\mathbf{B} = \begin{bmatrix} \mathbf{A'}_1 & & & & \\ & \mathbf{A'}_2 & & & \\ & & \mathbf{A'}_3 & & \\ & & & \ddots & \\ & & & & \mathbf{A'}_M \end{bmatrix},$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{A'}_1^I & & & & \\ & \mathbf{A'}_2^I & & & \\ & & \mathbf{A'}_3^I & & \\ & & & \ddots & \\ & & & & \mathbf{A'}_M^I \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 & \cdots & \mathbf{F}_M \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3 & \cdots & \mathbf{C}_M \end{bmatrix},$$

$$\mathbf{F}_i = \begin{bmatrix} -\mathbf{I}_{1,i}^{\phi} \\ -\mathbf{I}_{2,i}^{\phi} \\ \vdots \\ \mathbf{I}_{i,i}^{\phi} \\ \vdots \\ -\mathbf{I}_{M,i}^{\phi} \end{bmatrix},$$

$$\mathbf{C}_i = \begin{bmatrix} -\mathbf{I}^q_{1,i} \\ -\mathbf{I}^q_{2,i} \\ \vdots \\ -\mathbf{I}^q_{M,i} \end{bmatrix} \tag{13}$$

for $i = 1, 2, \ldots M$. We also define

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \\ \vdots \\ \boldsymbol{\lambda}_M \end{bmatrix},$$

$$\boldsymbol{\lambda}^I = \begin{bmatrix} \boldsymbol{\lambda}^I_{1,q} \\ \boldsymbol{\lambda}^I_{2,q} \\ \boldsymbol{\lambda}^I_{3,q} \\ \vdots \\ \boldsymbol{\lambda}^I_{M,q} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_M \end{bmatrix}.$$

For convenience, we use $\mathbf{B}_i$ and $\mathbf{E}_i$ to denote $\mathbf{A}'_i$ and $\mathbf{A}'^I_i$, respectively. With these definitions, Eq. (12) can be rewritten in the following form:

$$\mathbf{A}\boldsymbol{\lambda}' = \begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{F} & \mathbf{C} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\lambda}^I \end{Bmatrix} = \begin{Bmatrix} \mathbf{b} \\ \mathbf{0} \end{Bmatrix}. \tag{14}$$

Applying LDU decomposition [49] to matrix $\mathbf{A}$ yields:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{B} & \mathbf{E} \\ \mathbf{F} & \mathbf{C} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} \\ \mathbf{F}\mathbf{B}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{B} & \\ & \mathbf{S} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{B}^{-1}\mathbf{E} \\ & \mathbf{I} \end{bmatrix} \equiv \mathbf{L}_\mathbf{A}\mathbf{D}_\mathbf{A}\mathbf{U}_\mathbf{A} \end{aligned} \tag{15}$$

where $\mathbf{S}$ is the Schur complement matrix defined as

$$\mathbf{S} = \mathbf{C} - \mathbf{F}\mathbf{B}^{-1}\mathbf{E}. \tag{16}$$

Set $\lambda' = \mathbf{U}_\mathbf{A}^{-1}\mathbf{u}'$. By pre-multiplying $\mathbf{L}_\mathbf{A}^{-1}$ to Eq. (14), one obtains:

$$\mathbf{D}_\mathbf{A}\mathbf{u}' = \mathbf{L}_\mathbf{A}^{-1} \begin{Bmatrix} \mathbf{b} \\ \mathbf{0} \end{Bmatrix}, \quad \text{or} \quad \begin{bmatrix} \mathbf{B} & \\ & \mathbf{S} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{u}^I \end{Bmatrix} = \begin{Bmatrix} \mathbf{b} \\ -\mathbf{F}\mathbf{B}^{-1}\mathbf{b} \end{Bmatrix}, \tag{17}$$

where

$$\mathbf{u}' \equiv \begin{Bmatrix} \mathbf{u} \\ \mathbf{u}^I \end{Bmatrix}, \quad \mathbf{L}_\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{F}\mathbf{B}^{-1} & \mathbf{I} \end{bmatrix},$$

$$\mathbf{U}_\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{B}^{-1}\mathbf{E} \\ & \mathbf{I} \end{bmatrix}.$$

Since $\mathbf{D}_\mathbf{A}$ is a block-diagonal matrix, solving Eq. (17) is much easier than solving Eqs. (12) or (14). This is called Schur complement approach. The main difficult part of the solution is to find the inverse of $\mathbf{B}$ matrix, which will be addressed in the following. After Eq. (17) is solved for $\mathbf{y}$, the next step is to recover the original solution using the following equation:

$$\begin{aligned} \begin{Bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\lambda}^I \end{Bmatrix} &= \mathbf{U}_\mathbf{A}^{-1}\mathbf{u}' = \begin{bmatrix} \mathbf{I} & -\mathbf{B}^{-1}\mathbf{E} \\ & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{u}^I \end{Bmatrix} \\ &= \begin{Bmatrix} \mathbf{u} - \mathbf{B}^{-1}\mathbf{E}\mathbf{u}^I \\ \mathbf{u}^I \end{Bmatrix} \end{aligned} \tag{18}$$

It is worth of mentioning that $\mathbf{u}$ has already been determined when the right-hand-side of Eq. (17) is calculated. Therefore, we focus on solving equation:

$$\mathbf{S}\mathbf{u}^I = -\mathbf{F}\mathbf{B}^{-1}\mathbf{b} \tag{19}$$

Based on Eqs. (13) and (16), Eq. (19) is decomposed as:

$$\sum_{i=1}^{M} \mathbf{S}_i\mathbf{u}^I_i = \sum_{i=1}^{M} \left(\mathbf{C}_i - \mathbf{F}_i\mathbf{B}_i^{-1}\mathbf{E}_i\right)\mathbf{u}^I_i = -\sum_{i=1}^{M} \mathbf{F}_i\mathbf{B}_i^{-1}\mathbf{b}_i \tag{20}$$

Assume $\mathbf{B}_i\mathbf{w}_i = \mathbf{E}_i\mathbf{u}^I_i$. However, for domains without Dirichlet boundary conditions, the equation $\mathbf{B}_i\mathbf{w}_i = \mathbf{E}\mathbf{u}^I_i$ is not unique solvable. Langer et al. [35–37] proposed an approach to solve this problem. Assume that the subdomains $i = 1, \ldots, p$ are lack of the Dirichlet boundary conditions. For these subdomains, matrix $\mathbf{B}_i$ is modified as:

$$\mathbf{B}'_i = \mathbf{B}_i + \beta_i\mathbf{e}_i\mathbf{e}_i^T \tag{21}$$

where β_i is a positive constant and usually set to be 1, and $\mathbf{e}_i = (1, \ldots, 1)^T \in \mathrm{R}^{n_i}$. The corresponding Schur complement matrix is also modified as: $\mathbf{S}'_i = \mathbf{C}_i - \mathbf{F}_i\left(\mathbf{B}'_i\right)^{-1}\mathbf{E}_i$. Therefore, Eq. (19) is modified as:

$$\tilde{\mathbf{S}}\mathbf{u}^I = -\mathbf{F}\tilde{\mathbf{B}}^{-1}\mathbf{b} \tag{22}$$

where

$$\begin{aligned} \tilde{\mathbf{S}} &= \begin{bmatrix} \mathbf{S}'_1 \cdots \mathbf{S}'_p\ \mathbf{S}_{p+1} \cdots \mathbf{S}_M \end{bmatrix} \\ \tilde{\mathbf{B}} &= \begin{bmatrix} \mathbf{B}'_1 & & & & & \\ & \ddots & & & & \\ & & \mathbf{B}'_p & & & \\ & & & \mathbf{B}_{p+1} & & \\ & & & & \ddots & \\ & & & & & \mathbf{B}_M \end{bmatrix} \end{aligned} \tag{23}$$

With Eq. (21), $\mathbf{B}'_i\mathbf{w}_i = \mathbf{E}_i\mathbf{u}^I_i$ is unique solvable. After adding $\beta_i\mathbf{e}_i\mathbf{e}_i^T$ to matrix $\mathbf{B}_i$, we have:

$$\mathbf{B}'_i\mathbf{w}_i = \mathbf{B}_i\mathbf{w}_i + \beta_i\mathbf{e}_i\mathbf{e}_i^T \cdot \mathbf{w}_i = \mathbf{E}_i\mathbf{u}^I_i \quad .$$

To make sure that the solution vector $\mathbf{w}_i$ does not change after adding $\beta_i \mathbf{e}_i \mathbf{e}_i^T$, an additional orthogonal condition is needed: $\mathbf{e}_i^T \cdot \mathbf{w}_i = 0$. To satisfy this orthogonal condition, $\mathbf{w}_i$ is also modified as:

$$\mathbf{w}_i = \left(\mathbf{B'}_i\right)^{-1} \mathbf{E}_i \mathbf{u}_i^I + \gamma_i \mathbf{e}_i \quad for\ i = 1, 2, 3, \ldots, p \tag{24}$$

where γ_i is a parameter to be determined. Therefore, the left-hand-side of Eq. (22) becomes:

$$\begin{aligned}\tilde{\mathbf{S}}\mathbf{u}_i^I &= \sum_{i=1}^{p} \mathbf{C}_i \mathbf{u}_i^I - \mathbf{F}_i \left(\mathbf{B'}_i\right)^{-1} \mathbf{E}_i \mathbf{u}_i^I - \gamma_i \mathbf{F}_i \mathbf{e}_i \\ &\quad + \sum_{i=p+1}^{M} \mathbf{C}_i \mathbf{u}_i^I - \mathbf{F}_i \left(\mathbf{B}_i\right)^{-1} \mathbf{E}_i \mathbf{u}_i^I \\ &= \sum_{i=1}^{p} \mathbf{S'}_i \mathbf{u}_i^I - \gamma_i \mathbf{F}_i \mathbf{e}_i + \sum_{i=p+1}^{M} \mathbf{S}_i \mathbf{u}_i^I \\ &= \sum_{i=1}^{p} \mathbf{S'}_i \mathbf{u}_i^I + \sum_{i=p+1}^{M} \mathbf{S}_i \mathbf{u}_i^I - \sum_{i=1}^{p} \gamma_i \mathbf{G}_i \end{aligned} \tag{25}$$

where $\mathbf{G}_i$ is

$$\mathbf{G}_i = \mathbf{F}_i \mathbf{e}_i = \begin{bmatrix} -\mathbf{I}_{1,i}^{\phi} \mathbf{e}_i \\ -\mathbf{I}_{2,i}^{\phi} \mathbf{e}_i \\ \vdots \\ \mathbf{I}_{i,i}^{\phi} \mathbf{e}_i \\ \vdots \\ -\mathbf{I}_{M,i}^{\phi} \mathbf{e}_i \end{bmatrix} \tag{26}$$

Equation (25) can be re-written in matrix form as:

$$\tilde{\mathbf{S}}\mathbf{u}^I + \mathbf{G}\boldsymbol{\gamma} = -\mathbf{F}\tilde{\mathbf{B}}^{-1}\mathbf{b} \tag{27}$$

where

$$\mathbf{G} = \left[\, \mathbf{G}_1\ \mathbf{G}_2\ \mathbf{G}_3 \cdots \mathbf{G}_p \,\right]$$
$$\boldsymbol{\gamma} = \left[\, -\gamma_1 \cdots -\gamma_p \,\right]^T$$

In order to separate $\mathbf{u}^I$ and $\boldsymbol{\gamma}$, an orthogonal projection is defined as $\mathbf{P} = \mathbf{I} - \mathbf{G}\left(\mathbf{G^T G}\right)^{-1}\mathbf{G^T}$. By applying this approach, Eq. (27) becomes:

$$\mathbf{P}\tilde{\mathbf{S}}\mathbf{u}^I = -\mathbf{P}\mathbf{F}\tilde{\mathbf{B}}^{-1}\mathbf{b} \tag{28}$$

Once Eq. (28) is solved, the constant vector $\boldsymbol{\gamma}$ is recovered as:

$$\boldsymbol{\gamma} = \left(\mathbf{G^T G}\right)^{-1} \mathbf{G^T} \left(-\mathbf{F}\tilde{\mathbf{B}}^{-1}\mathbf{b} - \tilde{\mathbf{S}}\mathbf{u}^I\right) \tag{29}$$

The details of this approach is available in Ref. [35–37].

Calculating and storing matrix $\mathbf{P}$ explicitly may need long time and large computer memory. However, it is not necessary to do it because $\mathbf{G}_i$ is a sparse vector. Define V_i^A as a set which includes all the subdomains having interfaces with subdomain V_i. These subdomains are called "adjacent" subdomains herein. Therefore, $\mathbf{I}_{j,i}^{\phi}\mathbf{e}_i = \mathbf{0}$ for $\forall j \notin V_i^A$. If $j \in V_i^A$, the kth entry of $\mathbf{I}_{j,i}^{\phi}\mathbf{e}_i$ is:

$$\left(\mathbf{I}_{j,i}^{\phi}\mathbf{e}_i\right)_k = \begin{cases} 0 & k \notin E_{j,i}^A \\ 1 & k \in E_{j,i}^A \end{cases}$$

where $E_{j,i}^A$ is defined as a set which includes all the indexes of elements on S_{ij} for subdomain V_i. Therefore, we have:

$$\sum_{k=1}^{n_i} \left(\mathbf{I}_{j,i}^{\phi}\mathbf{e}_i\right)_k \cdot \left(\mathbf{I}_{j,i}^{\phi}\mathbf{e}_i\right)_k = n_i^j$$

where n_i^j is the total number of elements on S_{ij} for subdomain V_i.

The exact form of $\mathbf{G}_i^T\mathbf{G}_j$ is:

$$\begin{aligned}\mathbf{G}_i^T\mathbf{G}_j &= \sum_{k=1}^{M} (-1)^{\delta_{ki}+1} \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)^T \cdot (-1)^{\delta_{kj}+1} \left(\mathbf{I}_{k,j}^{\phi}\mathbf{e}_j\right) \\ &= \sum_{k=1}^{M} (-1)^{\delta_{ki}+1+\delta_{kj}+1} \sum_{l=1}^{n_k} \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)_l \cdot \left(\mathbf{I}_{k,j}^{\phi}\mathbf{e}_j\right)_l \end{aligned} \tag{30}$$

If $i = j$,

$$\begin{aligned}\mathbf{G}_i^T\mathbf{G}_i &= \sum_{k=1}^{M} (-1)^{2\delta_{ki}+2} \sum_{l=1}^{n_k} \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)_l \cdot \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)_l \\ &= \sum_{k=1}^{M} \sum_{l=1}^{n_k} \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)_l \cdot \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)_l = \sum_{k=1}^{M} n_i^k = 2n_i^I \end{aligned} \tag{31}$$

If $i \neq j$ and $j \in V_i^A$,

$$\begin{aligned}\mathbf{G}_i^T\mathbf{G}_j &= \sum_{k=1}^{M} (-1)^{\delta_{ki}+1} \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)^T \cdot (-1)^{\delta_{kj}+1} \left(\mathbf{I}_{k,j}^{\phi}\mathbf{e}_j\right) \\ &= \sum_{k=1}^{M} (-1)^{\delta_{ki}+1+\delta_{kj}+1} \sum_{l=1}^{n_k} \left(\mathbf{I}_{k,i}^{\phi}\mathbf{e}_i\right)_l \cdot \left(\mathbf{I}_{k,j}^{\phi}\mathbf{e}_j\right)_l \\ &= (-1)^{\delta_{ji}+1+\delta_{jj}+1} \sum_{l=1}^{n_j} \left(\mathbf{I}_{j,i}^{\phi}\mathbf{e}_i\right)_l \\ &\quad \cdot \left(\mathbf{I}_{j,j}^{\phi}\mathbf{e}_j\right)_l + (-1)^{\delta_{ii}+1+\delta_{ij}+1} \end{aligned}$$

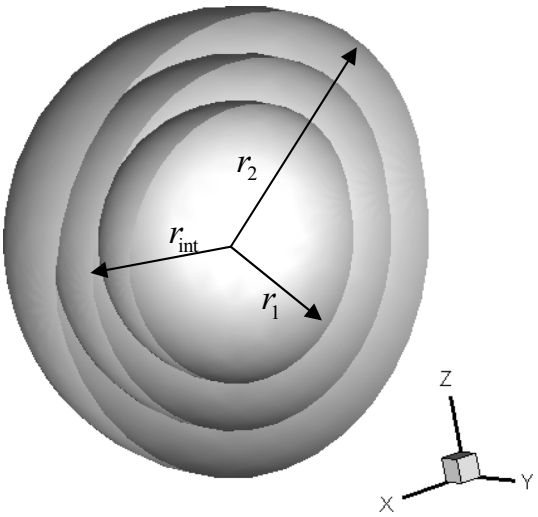

Fig. 2 Double shell model

$$\sum_{l=1}^{n_i} \left(\mathbf{I}_{i,i}^{\phi}\mathbf{e}_i\right)_l \cdot \left(\mathbf{I}_{i,j}^{\phi}\mathbf{e}_j\right)_l = -n_j^i - n_i^j = -2n_i^j \tag{32}$$

If $i \neq j$ and $j \notin V_i^A$, $\mathbf{G}_i^T\mathbf{G}_j = 0$. Eqs. (30)–(32) indicate that $\mathbf{G^T G}$ is a $p \times p$ sparse matrix and can be constructed directly by the number of elements on the interface. Usually, $\mathbf{G^T G}$ and its inverse are relatively small matrices. Therefore, instead of storing matrix $\mathbf{G}$, $\left(\mathbf{G^T G}\right)^{-1}$ is stored.

4 Algorithm

The overall algorithm starts with calculating the right-hand-side vector of Eq. (27) with fast multipole method. The next step is to call GMRES [50] to solve Eq. (28). For each iteration, the matrix-vector multiplication $\tilde{\mathbf{S}}\mathbf{z}^k \cdot \mathbf{z}^k$ is needed, where $\mathbf{z}^k$ is an input vector of kth iteration.

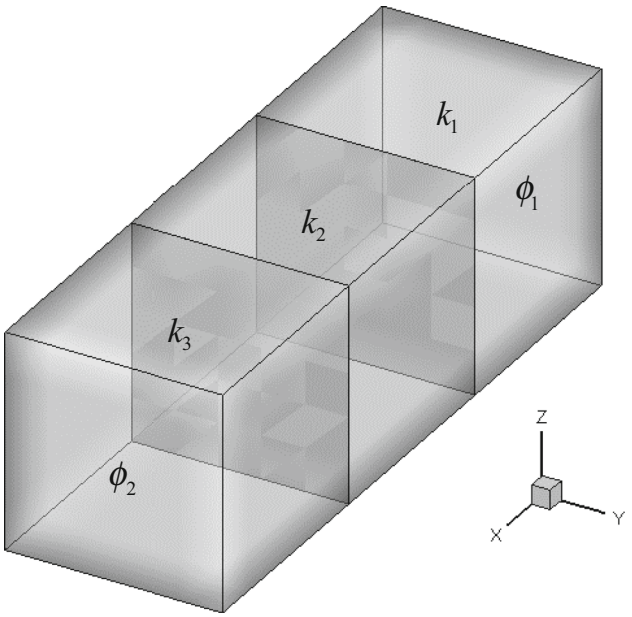

Fig. 3 Three block model

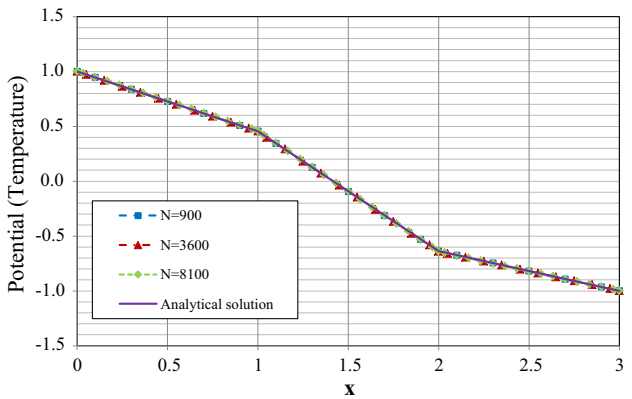

Fig. 4 Solutions of potential along the x direction of the three block model

Instead of calculating matrix $\tilde{\mathbf{S}}$ explicitly, we use fast multipole method to calculate matrix-vector product $\mathbf{Ez}^k$ first. After that, solve the equation $\tilde{\mathbf{B}}\mathbf{w}^k = \mathbf{Ez}^k$ for $\mathbf{w}^k$. Two different methods are used in this paper:

Table 1 Numerical results of the double shell model

No. of elements (N)	Heat flux on inner boundary		Temperature on interface		Heat flux on interface		Heat flux on outer boundary	
	Absolute value	Relative error (%)	Absolute value	Relative error (%)	Absolute value	Relative error (%)	Absolute value	Relative error (%)
360	−112.752	3.356	72.735	0.010	175.877	3.181	623.514	2.880
1584	−109.718	0.575	72.739	0.016	171.449	0.584	609.794	0.616
3672	−109.219	0.118	72.736	0.012	170.854	0.235	607.964	0.314
6624	−109.188	0.089	72.739	0.016	170.701	0.145	607.569	0.249
10,500	−109.086	0.004	72.747	0.028	170.627	0.101	607.698	0.270
Analytical solution	−109.091		72.727		170.454		606.0606	

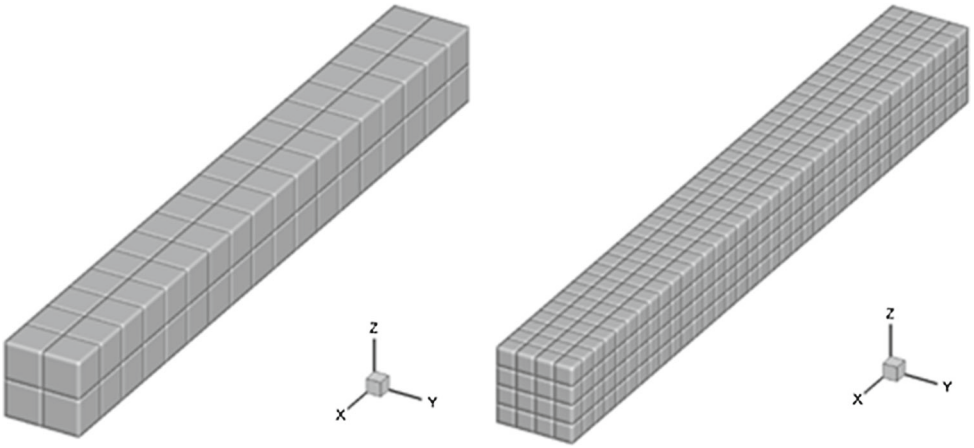

Fig. 5 Long-bar model divided into 72 and 576 subdomains

Table 2 Efficiency of different methods for long-bar models

Single domain FMM BEM			Solve Eq. (9) directly by using FMM			
N	Time	Iteration	*N*	*M*	Time	Iteration
7600	24s	18	21600	72	1m58s	129
30400	2m52s	19	86400	72	18m32s	324
121600	20m53s	34	172800	576	>1h	>500
486400	2h38m	55	691200	576	>4h	>500

New formulation-LU					New formulation-FMM				
N	*M*	Time	Memory (MB)	Iteration	*N*	*M*	Time	Memory (MB)	Iteration
21600	72	25s	1107	22	21600	72	9m2s	1067	22
86400	72	1m47s	1902	26	86400	72	1h15m	1114	26
172800	576	3m31s	1657	27	172800	576	1h19m	1229	27
691200	576	17m11s	8394	34	691200	576	8h20m	1956	37

(a) Calculate $\tilde{\mathbf{B}}_i$ by using the conventional approach. Use LU decomposition to solve $\tilde{\mathbf{B}}_i \mathbf{w}_i^k = \mathbf{E}_i \mathbf{z}_i^k$. This can be done effectively, if the size of $\tilde{\mathbf{B}}_i$ is moderate (e.g., less than ten thousands for current PCs). This approach is called as "New Formulation-LU".

(b) Use fast multipole method to solve $\tilde{\mathbf{B}}_i \mathbf{w}_i^k = \mathbf{E}_i \mathbf{z}_i^k$ iteratively, if the size of $\tilde{\mathbf{B}}_i$ is large. This approach is called as "New Formulation-FMM".

The next step is to substitute $\mathbf{w}^k = \tilde{\mathbf{B}}^{-1}\mathbf{E}\mathbf{z}^k$ into the expression of $\mathbf{S}$ (Eq. (16)) and complete the matrix-vector

Table 3 Errors of long-bar models with new formulation-LU

N	M	Error (%)	
		ϕ	q
21,600	72	0.131	6.579
86,400	72	0.066	4.965
172,800	576	0.118	6.797
691,200	576	0.061	5.628

product as $\mathbf{Sz}^k = \mathbf{Cz}^k - \mathbf{Fw}^k$. Note that $\mathbf{C}$ and $\mathbf{F}$ are sparse matrices. Therefore, both $\mathbf{C}$ and $\mathbf{F}$ are not stored explicitly. Only the locations of nonzero elements are stored. The last step of each iteration is to apply the projection $\mathbf{P}$ for those subdomains without Dirichlet boundary conditions. After Eq. (28) is solved, the original solution is obtained as $\lambda' = \mathbf{U}_\mathbf{A}^{-1}\mathbf{u}'$. For the subdomain without Dirichlet boundary condition, calculate the constant $\boldsymbol{\gamma}$ and recover the original solution.

5 Numerical results

Several numerical examples are given to show the accuracy and efficiency of the new formulation of domain decomposition or multidomain BEM derived in this paper. For the first two cases, only LU decomposition is used to solve $\tilde{\mathbf{B}}\mathbf{w}^k = \mathbf{Ez}^k$. Triangular constant elements are used in the discretization and all integrals are done analytically. The tolerance for convergence of the GMRES iterative solver is set at 1.0E−6. The numbers of CPU cores used in the first two examples are equal to the numbers of subdomains of the corresponding models. For the long-bar model and Menger sponger model, 12 CPU cores are used. For the single fuel cell model, 12 CPU cores are used as well. For the model with 25 fuel cells, 32 CPU cores are used.

5.1 A double shell model

The first case is a double-shell model. Two shells with different heat conduction coefficients are combined together. There are two subdomains in this model. The inner boundary of the outer shell attaches to the outer boundary of the inner shell. Figure 2 shows the geometry of the model. ϕ_1, q_1 and ϕ_2, q_2 are the temperature and normal heat flux on the inner and outer boundaries, respectively. k_1 and k_2 are the heat conduction coefficients of the inner and outer shells, respectively. r_1 and r_2 are the radii of inner and outer boundaries of the model. r_{int} is the radius of the interface. This is an axisymmetric problem and the analytical solution of the temperature field inside the shells can be found to be:

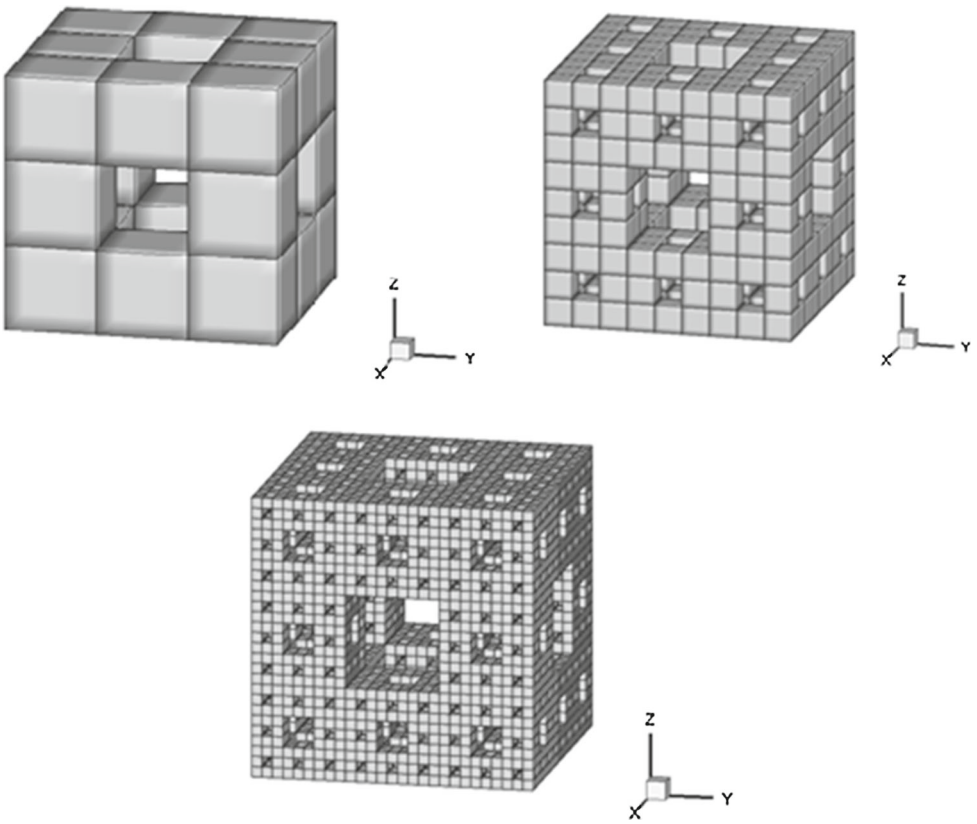

Fig. 6 Menger sponge model divided into 20, 400 and 8000 subdomains

Table 4 Efficiency of different methods for solving Menger sponge models

Single domain FMM BEM			Solve Eq. (9) directly by using FMM			
N	Time	Iteration	N	M	Time	Iteration
3600	28s	28	6000	20	21s	59
14400	2m3s	26	24000	20	1m37s	75
52800	16m33s	50	120000	400	8m24s	85
211200	1h41m	85	480000	400	40m52s	105
			2400000	8000	2h43m	98

New formulation-LU					New formulation-FMM				
N	M	Time	Memory (MB)	Iteration	N	M	Time	Memory (MB)	Iteration
6000	20	10s	1093	19	6000	20	1m43s	1047	19
24000	20	38s	1308	26	24000	20	9m52s	1131	25
120000	400	3m23s	1452	34	120000	400	45m55s	1151	34
480000	400	15m26s	5760	44	480000	400	4h24m	1471	45
2400000	8000	1h03m	9563	41	2400000	8000	>12h		-

$$\phi(r) = \begin{cases} \frac{A_1}{r} + B_1 & r_1 \le r \le r_{int} \\ \frac{A_2}{r} + B_2 & r_{int} \le r \le r_2 \end{cases} \tag{33}$$

where

$$A_1 = \frac{\phi_1 - \phi_2}{\left(\left(\frac{1}{r_1} - \frac{1}{r_{int}}\right) - \frac{k_1}{k_2}\left(\frac{1}{r_2} - \frac{1}{r_{int}}\right)\right)},$$

$$A_2 = \frac{\phi_1 - \phi_2}{\left(\frac{k_2}{k_1}\left(\frac{1}{r_1} - \frac{1}{r_{int}}\right) - \left(\frac{1}{r_2} - \frac{1}{r_{int}}\right)\right)},$$

$$B_1 = \phi_1 - \frac{A_1}{r_1}, \quad B_2 = \phi_2 - \frac{A_2}{r_2}.$$

From Eq. (33), we obtain the analytical solution of the heat flux along the radial direction:

$$q = \begin{cases} -k_1 \frac{A_1}{r^2} & r_1 \le r \le r_{int} \\ -k_2 \frac{A_2}{r^2} & r_{int} \le r \le r_2 \end{cases}. \tag{34}$$

Assume $r_1 = 0.3$, $r_2 = 0.5$, and $r_{int} = 0.4$, and the heat conduction coefficients $k_1 = 1.0$ and $k_2 = 0.5$. The temperature on the inner and outer boundary are given as: $\phi_1 = 50$ and $\phi_2 = 100$.

Table 1 shows the numerical results of the heat flux and temperature on the interface and boundary. It can be seen that the numerical results converge to the analytical solution as the number of elements of the BEM model increases. This test shows that the multidomain BEM is effective and accurate.

5.2 A block model

The next example is three connected blocks with different heat conduction coefficients (Fig. 3). There are three subdomains in this model. We assume $k_1 = 1.0$, $k_2 = 0.5$ and $k_3 = 1.5$. The boundary conditions applied are: $\phi_1 = 1$ and $\phi_2 = -1$ at the two ends. Figure 4 shows that the analytical solution and numerical results of the potential along the x

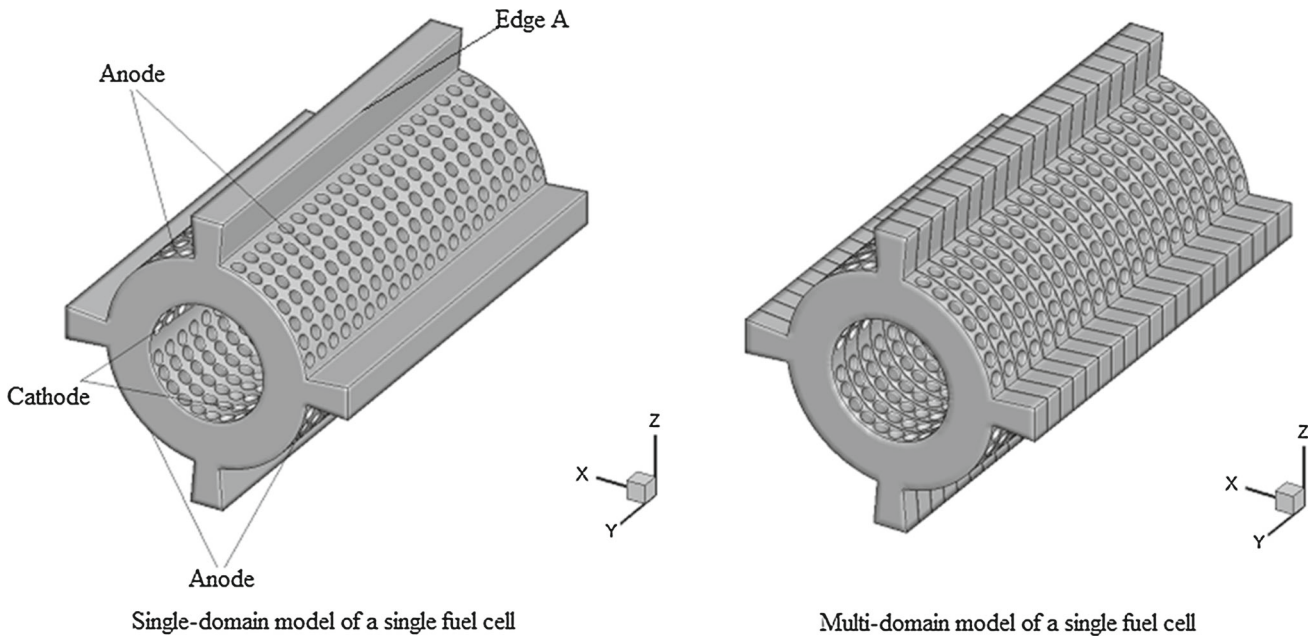

Fig. 7 Single-domain and multi-domain models of a single solid oxide fuel cell

direction match very well with each other for different number of elements (N) used. This further verifies the proposed multidomain BEM formulation and its implementation.

5.3 A long-bar model

Next, the efficiency of the new proposed formulation is compared with the traditional fast multipole BEM for single-domain problems. Figure 5 shows a long-bar model divided into 72 and 576 subdomains. All the subdomains have the same material property. The boundary condition is also given as: $\phi_1 = 1$ and $\phi_2 = -1$ at the two ends as shown in Fig. 5.

Table 2 shows the elapsed wall-clock time and the number of iterations for different approaches. It can be seen that solving Eq. (9) directly using fast multipole method needs most iterations to obtain the solution. With Schur complement approach, the number of iterations reduce significantly and is smaller than that of the traditional single-domain fast multipole BEM for most cases. When we compare the efficiency of the two different approaches of solving $\mathbf{Bw}^k = \mathbf{Ez}^k$, using fast multipole method takes much longer time than using the LU decomposition. The reason is that fast multipole method needs some iterations to solve $\mathbf{Bw}^k = \mathbf{Ez}^k$ for every matrix-vector multiplication $\mathbf{Sz}^k$. However, for LU decomposition, the matrix $\mathbf{B}$ only needs to be decomposed once.

Table 3 shows the error of the numerical results of the new proposed formulation with LU decomposition. The error is calculated as:

$$error = \sqrt{\frac{\|\mathbf{u}_{num} - \mathbf{u}_{exact}\|^2}{\|\mathbf{u}_{exact}\|^2}}.$$

For all cases, the errors of ϕ are smaller than 1 % and the errors of q are close to 5 %, which demonstrates the accuracy of the new proposed formulation. It is worth to note that the error of q is higher than that of ϕ. This is the issue of using constant elements because constant elements cannot give the results of q accurately around the corners and edges of a model.

5.4 Menger sponge models

In this example, the new formulation of the multidomain fast multipole BEM is used to solve the potential problem of a Menger sponge model (Fig. 6). The sponger model is divided into 20, 400 and 8000 subdomains. Each subdomain has the same property. The boundary condition is given by imposing $\phi = x + y + z$ on all boundary surfaces and the normal derivative of potential q is sought. The efficiencies of the four different approaches are compared in

Table 4, which shows that using Eq. (9) directly to solve the problem still needs most iterations and Schur complement reduces the number of iterations significantly. The approach with LU decomposition still takes less time but needs much larger memory storage than the one with fast multipole method. The largest BEM model solved has 2.4 million unknowns and solved in 1 h 3 min on a multicore CPU computer at Ohio Supercomputer Center.

5.5 Solid oxide fuel cell models

In this example, the new formulation of the multidomain fast multipole BEM is used to solve large-scale heat conduction

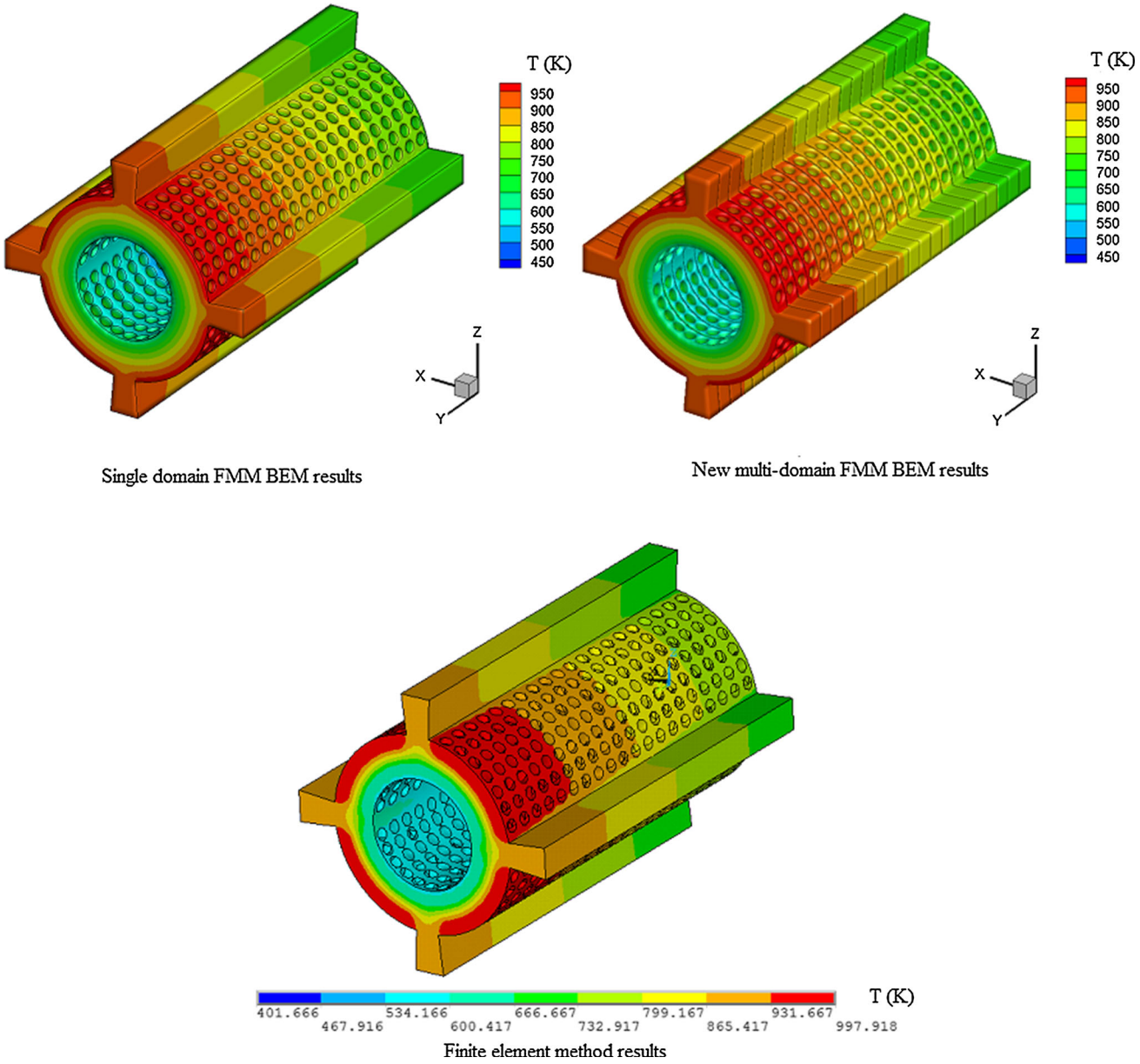

Fig. 8 Temperature distributions of the single fuel cell models with three different methods

problems of the solid oxide fuel cells. The geometry of a single cell is shown on the left of Fig. 7. The length of the single cell is 0.3 m. The diameters of anode and cathode are 0.15 and 0.084 m, respectively. Both the width and height of the single cell are 0.21 m. Notice that there are 1000 side holes on both the inner and outer surfaces of the cell, which makes the modeling especially challenging for both the BEM and FEM.

First, the heat conduction problem of a single fuel cell is calculated by using the new multidomain fast multipole BEM, conventional single-domain fast multipole BEM and finite element method (ANSYS®). The multidomain model of a single fuel cell is created by dividing the cell into 20 subdomains along Y axis, which is shown on the right of Fig. 7. Each subdomain has 13,534 elements. The single-domain BEM model has a total of 214,136 elements. The finite element model has a total of 365,339 linear tetrahedron elements. The boundary conditions of the model are given as: linearly distributed temperature on the anode from the near end to the far end with the range of 1000–750 K; linearly distributed temperature on the cathode from the near end to the far end with the range of 600–400 K; zero heat flux on all the other surfaces. The heat conduction coefficient is 90.5 W/(m.K).

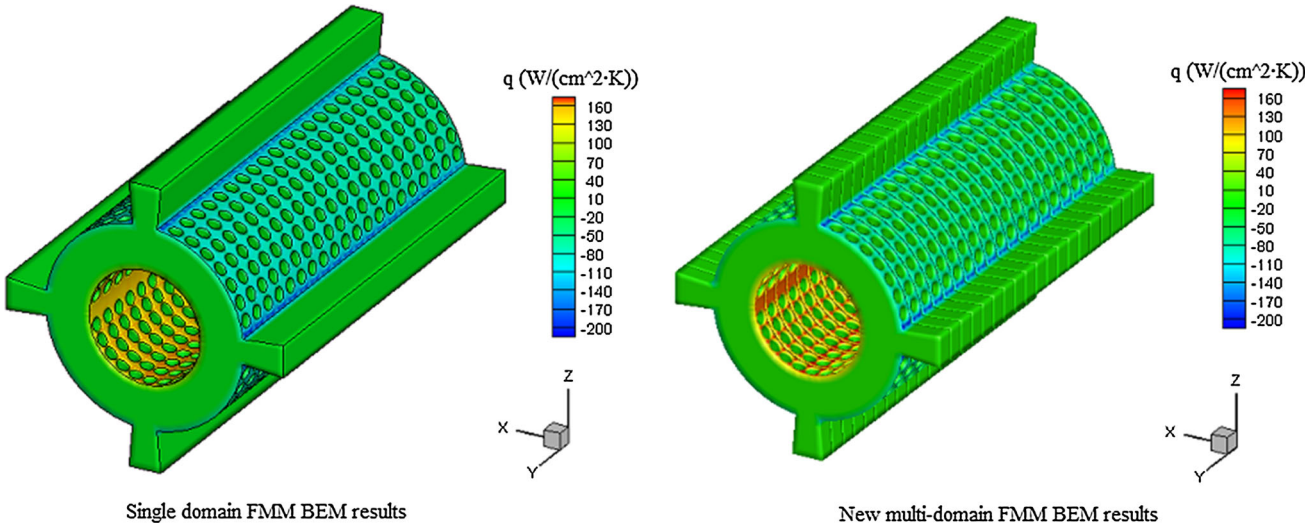

Fig. 9 Heat flux distributions of the single fuel cell models with two the different methods

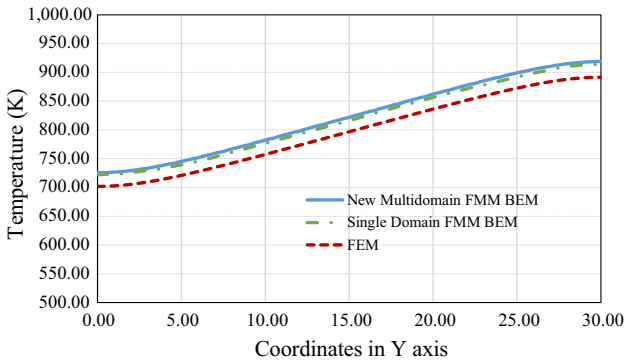

Fig. 10 Temperature distribution along Edge A

Figures 8 and 9 show the contour plots of the computed temperature and heat flux distributions on the surfaces of the cell. Figure 10 shows the computed temperature variations along Edge A (the narrow top surface, shown in Fig. 7). From Figures 8 and 10, it is observed that the temperature distributions obtained by the three different methods agree very well to each other. From Fig. 9, it can be seen that the heat flux distributions obtained by the single-domain FMM BEM and new multi-domain FMM BEM are also very close to each other. In Fig. 10, the difference between the results of the new multidomain fast multipole BEM and the results of the conventional single-domain fast multipole BEM is within 1 %. The difference between the results of the new multidomain fast multipole BEM and the results of the FEM is within 3 %. The difference in the results of the new multidomain BEM and the single domain BEM is most likely caused by the additional elements or discretization introduced on the interfaces. The difference in the results of the two BEM approaches and the FEM (Fig. 10) is most likely a convergence issue. Due to the limitation of the ANSYS® academic license and the PC hardware, further refined FEM models were not attempted.

Next, a larger BEM model with 25 fuel cells is calculated by using the new multidomain fast multipole BEM. The boundary conditions are the same as the single fuel cell model. The total number of elements (DOFs) is 6,767,000. The entire model is divided into 500 subdomains and each subdomain has 13,534 elements. LU decomposition is used for solving $\mathbf{Bw}^k = \mathbf{Ez}^k$. The model is solved on a cluster with 32 cores at the Ohio Supercomputer Center and it took about 2 h 56 min to solve the BEM model using the new multidomain fast multipole BEM (with 43 iterations and a tolerance of 10E−4). The temperature distribution of the model are shown in Fig. 11. The conventional single-domain fast multipole BEM was also attempted in solving this large model. However, after using 96 CPU hours on the same cluster, it was terminated by the system and failed to deliver the results. The iteration number is 43 and residual is still at 0.25E−2 when the job was stopped. Due to limitations on requesting longer CPU times on the cluster, no further run was attempted. This partially verified the fact the new multidomain BEM approach proposed in this paper can significantly improve the conditioning of the system of equations and thus can significantly accelerate the solutions of large-scale BEM models.

6 Discussions

In this paper, a new formulation of multidomain fast multipole BEM is proposed. Unlike the traditional BEM for single-domain problems, the multidomain BEM decomposes the entire domain into some subdomains. Additional inter-

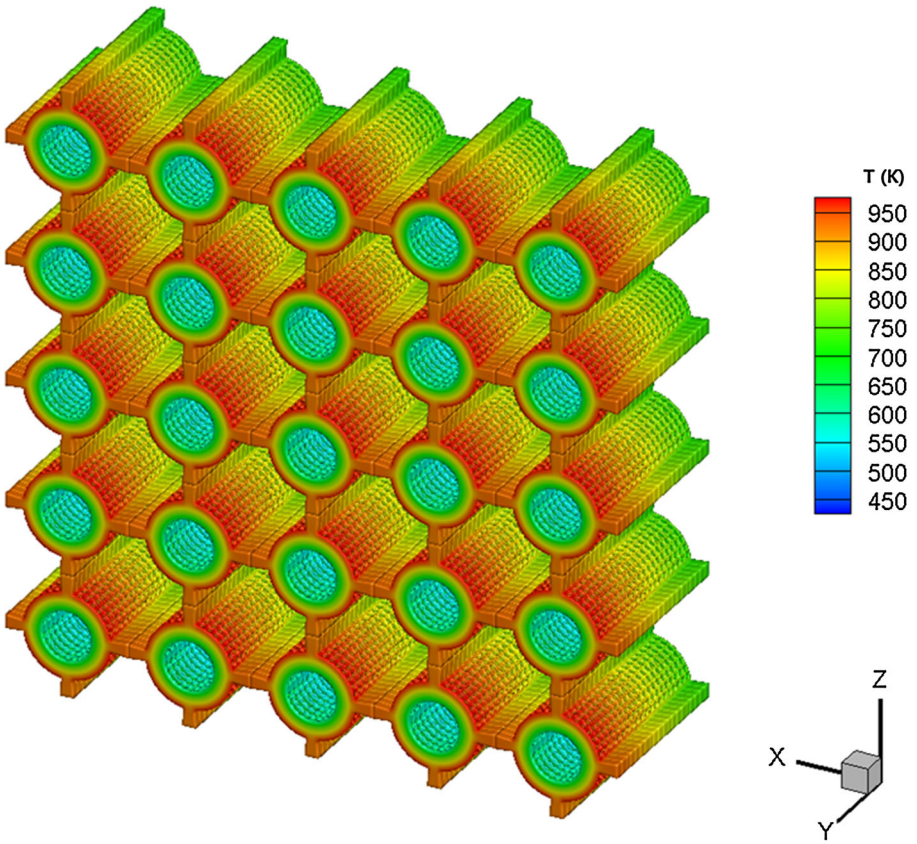

Fig. 11 Temperature distribution on surfaces of the 5×5 fuel cell model (with a total number of 6,767,000 elements)

face conditions are added to relate the variables on the interface. This approach increases the total number of elements and the coefficient matrix becomes larger (See Eq. (9)). Instead of solving the large linear system of equations directly, we partition and decompose the entire coefficient matrix. By eliminating part of the unknown variables, solving the original entire linear system turns out to be equivalent to solving a smaller linear system with Schur complement.

The numerical results show that, the new multidomain fast multipole BEM is accurate and solutions with iterative solvers can be obtained using fewer iterations compared to solving Eq. (9) directly. Compared to the traditional single-domain fast multipole BEM, the new multidomain fast multipole BEM also uses fewer iterations for most cases. Based on the derivations shown in this paper, the new formulation is suitable for parallel computing, which reduces the communication between different subdomains to the minimum. It is demonstrated that the new multidomain fast multipole BEM is capable of solving large-scale problems with complex geometries efficiently.

The key step to achieve the efficiency of the new formulation is how to solve $\tilde{\mathbf{B}}\mathbf{w}^k = \mathbf{E}\mathbf{z}^k$. In this study, we use two different methods. One is to use conventional BEM to calculate $\tilde{\mathbf{B}}$ and store it. The LU decomposition is called later to solve $\tilde{\mathbf{B}}\mathbf{w}^k = \mathbf{E}\mathbf{z}^k$. Another approach is to apply fast multipole method to solve it. However, both methods are not perfect. With the first approach, the new multidomain fast multipole BEM can solve the problem quickly. However, it requires large amount of computer memory because it requires storing the matrix $\tilde{\mathbf{B}}$ of each subdomain. The second approach does not need large amount of computer memory. However, it takes longer time to obtain the solution due to the iterative solution process. A potential and new approach is to use a fast direct solver which does not need to store the entire coefficient matrix or obtain the solution iteratively. Another possible approach to improve the efficiency of the new multidomain fast multipole method is to improve the conditioning of matrix $\tilde{\mathbf{S}}$ by using the re-numbering technique in Ref. [48] or a more efficient preconditioner. Extensions of the new multidomain fast multipole BEM to solving large-scale elasticity, acoustic and elastodynamic problems will be straightforward and can potentially lead the BEM to interesting areas of new applications.

Acknowledgments The authors would like to acknowledge the support of Ohio Supercomputer Center (OSC). All the computing jobs for the examples were done on the "Oakley" cluster at OSC.

References

1. Kompiš V (2002) Selected topics in boundary integral formulations for solids and fluids., CISM courses and lectures/International Centre for Mechanical SciencesSpringer, New York
2. Gao XW, Wang J (2009) Interface integral BEM for solving multi-medium heat conduction problems. Eng Anal Bound Elem 33(4):539–546. doi:10.1016/j.enganabound.2008.08.009
3. Du F, Lovell MR, Wu TW (2001) Boundary element method analysis of temperature fields in coated cutting tools. Int J Solids Struct 38(26–27):4557–4570. doi:10.1016/S0020-7683(00)00291-2
4. Florez WF, Power H, Chejne F (2002) Numerical solution of thermal convection problems using the multidomain boundary element method. Numer Methods Partial Differ Equ 18(4):469–489. doi:10.1002/num.10016
5. Yu W, Wang Z, Hong X (2004) Preconditioned multi-zone boundary element analysis for fast 3D electric simulation. Eng Anal Bound Elem 28(9):1035–1044. doi:10.1016/j.enganabound.2004.02.006
6. Ramšak M, Škerget L (2007) 3D multidomain BEM for solving the Laplace equation. Eng Anal Bound Elem 31(6):528–538. doi:10.1016/j.enganabound.2006.10.006
7. Zhang J (2008) Multidomain thermal simulation of CNT composites by hybrid BNM. Numer Heat Transf Part B: Fundam 53(3):246–258. doi:10.1080/10407790701790176
8. Liu YJ (2008) A fast multipole boundary element method for 2D multi-domain elastostatic problems based on a dual BIE formulation. Comput Mech 42(5):761–773. doi:10.1007/s00466-008-0274-2
9. Wang H, Yao Z, Wang P (2005) On the preconditioners for fast multipole boundary element methods for 2D multi-domain elastostatics. Eng Anal Bound Elem 29(7):673–688. doi:10.1016/j.enganabound.2005.03.002
10. Gao XW, Guo L, Zhang C (2007) Three-step multi-domain BEM solver for nonhomogeneous material problems. Eng Anal Bound Elem 31(12):965–973. doi:10.1016/j.enganabound.2007.06.002
11. Gao XW, Yang K (2009) Interface integral BEM for solving multi-medium elasticity problems. Comput Methods Appl Mech Eng 198(15–16):1429–1436. doi:10.1016/j.cma.2008.12.007
12. Layton JB, Ganguly S, Balakrishna C, Kane JH (1997) A symmetric Galerkin multi-zone boundary element formulation. Int J Numer Methods Eng 40:2913–2931
13. Gao XW, Davies TG (2000) 3D multi-region BEM with corners and edges. Int J Solids Struct 37(11):1549–1560. doi:10.1016/S0020-7683(98)00276-5
14. Luo JF, Liu YJ, Berger EJ (2000) Interfacial stress analysis for multi-coating systems using an advanced boundary element method. Comput Mech 24(6):448–455. doi:10.1007/s004660050004
15. Zhang J, Zhuang C, Qin X, Li G, Sheng X (2010) FMM-accelerated hybrid boundary node method for multi-domain problems. Eng Anal Bound Elem 34(5):433–439. doi:10.1016/j.enganabound.2009.12.005
16. Wu TW (2008) Multi-domain boundary element method in acoustics. In: Marburg S, Nolte B (eds) Computational acoustics of noise propagation in fluids—finite and boundary element methods. Springer, Berlin, pp 367–386. doi:10.1007/978-3-540-77448-8_14
17. Kallivokas LF, Juneja T, Bielak J (2005) A symmetric Galerkin BEM variational framework for multi-domain interface problems. Comput Method Appl Mech Eng 194(34–35):3607–3636. doi:10.1016/j.cma.2004.07.034
18. Sarradj E (2003) Multi-domain boundary element method for sound fields in and around porous absorbers. Acta Acust United Acust 89(1):21–27
19. Cheng CYR, Seybert AF, Wu TW (1991) A multidomain boundary element solution for silencer and muffler performance prediction. J Sound Vib 151(1):119–129. doi:10.1016/0022-460X(91)90655-4
20. Wu HJ, Liu YJ, Jiang WK (2012) A fast multipole boundary element method for 3D multi-domain acoustic scattering problems based on the Burton-Miller formulation. Eng Anal Bound Elem 36(5):779–788. doi:10.1016/j.enganabound.2011.11.018
21. Chaillat S, Bonnet M, Semblat JF (2009) A new fast multi-domain BEM to model seismic wave propagation and amplification in 3-D geological structures. Geophys J Int 177(2):509–531. doi:10.1111/j.1365-246X.2008.04041.x
22. Ahmad S, Banerjee PK (1988) Multi-domain BEM for two-dimensional problems of elastodynamics. Int J Numer Methods Eng 26:891–911
23. Maier G, Diligenti M, Carini A (1991) A variational approach to boundary element elastodynamic analysis and extension to multidomain problems. Comput Method Appl Mech Eng 92(2):193–213. doi:10.1016/0045-7825(91)90239-3
24. Ramšak M, Škerget L (2005) A multidomain boundary element method for two equation turbulence models. Eng Anal Bound Elem 29(12):1086–1103. doi:10.1016/j.enganabound.2005.07.007
25. Ramšak M, Škerget L, Hriberšek M, Žunič Z (2005) A multidomain boundary element method for unsteady laminar flow using stream function-vorticity equations. Eng Anal Bound Elem 29(1):1–14. doi:10.1016/j.enganabound.2004.09.002
26. Florez WF, Power H (2001) Comparison between continuous and discontinuous boundary elements in the multidomain dual reciprocity method for the solution of the two-dimensional Navier-Stokes equations. Eng Anal Bound Elem 25(1):57–69. doi:10.1016/S0955-7997(00)00051-5
27. Brebbia C, Power H, Škerget L (2007) Domain decomposition techniques for boundary elements: application to fluid flow. WIT Press, Southampton
28. Chen T, Wang B, Cen Z, Wu Z (1999) A symmetric Galerkin multi-zone boundary element method for cohesive crack growth. Eng Fract Mech 63:591–609
29. Doblare M, Espiga F, Gracia L, Alcantud M (1990) Study of crack propagation in orthotropic materials by using the boundary element method. Eng Fract Mech 37(5):953–967. doi:10.1016/0013-7944(90)90020-H
30. DavıI G, Milazzo A (2001) Multidomain boundary integral formulation for piezoelectric materials fracture mechanics. Int J Solids Struct 38(40–41):7065–7078. doi:10.1016/S0020-7683(00)00416-9
31. Kane JH, Kashava Kumar BL, Saigal S (1990) An arbitrary condensing, noncondensing solution strategy for large scale, multi-zone boundary element analysis. Comput Method Appl Mech Eng 79(2):219–244. doi:10.1016/0045-7825(90)90133-7
32. Rigby RH, Aliabadi MH (1995) Out-of-core solver for large, multi-zone boundary element matrices. Int J Numer Methods Eng 38(9):1507–1533. doi:10.1002/nme.1620380905
33. Hsiao GC, Wendland W (1991) Domain decomposition in boundary element method. Paper presented at the fourth international symposium on domain decomposition methods for partial differential equations, Philadelphia
34. Hsiao GC, Steinbach O, Wendland WL (2000) Domain decomposition methods via boundary integral equations. J Comput Appl Math 125:521–537
35. Langer U, Of G, Steinbach O, Zulehner W (2007) Inexact fast multipole boundary element tearing and interconnecting methods. SIAM J Sci Comput 29:290–314
36. Of G, Steinbach O (2009) The all-floating boundary element tearing and interconnecting method. J Numer Math 17(4):277–298. doi:10.1515/JNUM.2009.014

37. Langer U, Steinbach O (2003) Boundary element tearing and interconnecting methods. Computing 71(3):205–228. doi:10.1007/s00607-003-0018-2
38. Rokhlin V (1985) Rapid solution of integral equations of classical potential theory. J Comp Phys 60:187–207
39. Greengard LF, Rokhlin V (1987) A fast algorithm for particle simulations. J Comput Phys 73(2):325–348
40. Greengard LF (1988) The rapid evaluation of potential fields in particle systems. The MIT Press, Cambridge
41. Nishimura N (2002) Fast multipole accelerated boundary integral equation methods. Appl Mech Rev 55(4 (July)):299–324
42. Liu YJ, Nishimura N (2006) The fast multipole boundary element method for potential problems: a tutorial. Eng Anal Bound Elem 30(5):371–381
43. Liu YJ (2009) Fast multipole boundary element method—theory and applications in engineering. Cambridge University Press, Cambridge
44. Pechstein C (2009) Boundary element tearing and interconnecting methods in unbounded domains. Appl Numer Math 59(11):2824–2842. doi:10.1016/j.apnum.2008.12.031
45. Ingber MS, Tanski JA, Alsing P (2007) A domain decomposition tool for boundary element methods. Eng Anal Bound Elem 31(11):890–896. doi:10.1016/j.enganabound.2007.03.002
46. Bui TT, Popov V (2009) Domain decomposition boundary element method with overlapping sub-domains. Eng Anal Bound Elem 33(4):456–466. doi:10.1016/j.enganabound.2008.09.002
47. Bebendorf M (2000) Approximation of boundary element matrices. Numer Math 86:565–589
48. Zhang J, Lu C, Li Y, Han L, Wang P, Li G (2014) A domain renumbering algorithm for multi-domain boundary face method. Eng Anal Bound Elem 44:19–27. doi:10.1016/j.enganabound.2014.04.009
49. Chen K (2005) Matrix preconditioning techniques and applications. Cambridge University Press, Cambridge
50. Saad Y (2003) Iterative methods for sparse linear system, 2nd edn. The Society for Industrial and Applied Mathematics, Philadelphia