IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 25, 2018, accepted January 23, 2019, date of publication February 8, 2019, date of current version February 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896615

Simulation of Challenging Electromagnetic Problems Using a Massively Parallel Finite Element Method Solver

**SHENG ZUO, DANIEL GARCÍA DOÑORO, (Member, IEEE),
YU ZHANG, YANG BAI, AND XUNWANG ZHAO**
Shaanxi Key Laboratory of Large Scale Electromagnetic Computing, Xidian University, Xi'an 710071, China

Corresponding author: Yu Zhang (yuzhang@mail.xidian.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFE0121600, in part by the China Postdoctoral Science Foundation funded project under Grant 2017M613068, and in part by the National Key Research and Development Program of China under Grant 2017YFB0202102.

ABSTRACT This communication presents an efficient massively parallel finite element method solver for the solution of complex and electrically large electromagnetic problems with arbitrary structures. The solver makes use of a domain decomposition algorithm to decompose the original problem into several non-overlapping sub-domains that may be solved independently in parallel through the application of the corresponding transmission conditions on the interfaces of the adjacent sub-domains. A numerical exact mesh truncation algorithm called finite element-iterative integral equation evaluation, accelerated with multilevel fast multipole algorithm, is implemented to meet the highly accurate requirements of today's challenging simulations. What's more, a hybrid message passing interface and an open multi-processing parallel framework are designed to achieve large-scale parallel performance on supercomputers. Through several numerical examples, the accuracy, effectiveness, and scalability of the proposed solver will be demonstrated, achieving more than 60% parallel efficiency on an eight times CPU core scale (from 1280 to 10 240 cores).

INDEX TERMS Finite element method (FEM), domain decomposition method (DDM), large-scale parallel computing, finite element-iterative integral equation evaluation (FE-IIEE), ten thousand CPU cores.

I. INTRODUCTION

The finite element method (FEM) is a numerical technique widely used in predicting the behavior of radiating and scattering objects of an arbitrary shape for its ability to handle complex geometrical features and material properties. Unfortunately, the direct application of the FEM in the analysis of complex and electrically large electromagnetic problems leads to solve huge, highly ill-conditioning and, even possibly indefinite, complex sparse matrix equation. In this scenario, the use of memory efficient iterative solvers seems the appropriate choice, however they may experience slow convergence rates or even divergence when solving challenging electromagnetics problems [1]. Contrary, direct solvers such as MUMPS [2] or MKL PARDISO [3] are very reliable but, computationally speaking, highly demanding for large-scale parallel computing. On the other hand, domain decomposition methods (DDMs) [4] provide an efficient alternative to overcome the aforementioned problems. DDMs are based on the idea of *"divide and conquer"*, that is, the original and hard to solve electromagnetic problem is divided into smaller and easier to handle independent sub-problems amenable to solve by using sparse direct solvers through the imposition of appropriate transmission conditions (TCs) on the interfaces of adjacent sub-domains.

In the past decades, the theory of DDM has developed rapidly, and the capability of FEM for solving complex and electrically large problems has been dramatically enhanced [5]–[12]. However, most of these works were focused on the improvement of the TC [5], [11], [12] to obtain stable, optimal and convergent algorithms, frequently solving models with characteristics of periodic structures

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

such as photonic band gap (PBG), electromagnetic band gap (EBG) or finite antenna array and so on [6]–[8]. In recent works authors have performed large-scale electromagnetic simulations using high-performance computing schemes together with DDM algorithms as in [13]. However, their computational domain discretization and mesh truncation was made using first order basis functions and first order absorbing boundary condition (ABC), respectively. Thus, it becomes really hard to meet the highly accurate requirement of nowadays challenging electromagnetic simulations. Jia *et al.* [14] introduced the hierarchical matrix technique to improve the efficiency of the finite element method (FEM) – boundary element method (BEM) – DDMs. Although the BEM provides a very accurate mesh truncation boundary, its massively parallel implementation is not straightforward due to the FEM matrix shape. At the same time, these works are still focused on the solution of periodic structures without too effective improvement of the FEM's computing ability for arbitrary complex structures. Consequently, from the current point of view, compared to extremely efficient parallel integral equation works, as in [15] and [16], there is lack of feasible and effective parallel FEM general solvers for solving complex and electrically large aperiodic structures in practical engineering.

In this paper, a massively parallel FEM solver for solving complex and electrically large aperiodic problems is presented. The solver implements a first order non-overlapping DDM technology and a hybrid parallel paradigm that combines a distributed-memory scheme by using message passing interface (MPI) [17] and a shared-memory scheme by using Open Multiprocessing (OpenMP) [18]. This combination provides a massively efficient parallel implementation, that allows us to use all the computing power and storage capacity of large high-performance computing (HPC) clusters and supercomputers.

Furthermore, the FEM mesh truncation technology for open problems is of capital importance in full-wave electromagnetic simulations when modeling radiating devices (antennas) and scattering open problems (for example, prediction of radar cross section), or the spurious radiation in guiding structures or, in general, passive devices. When implementing a FEM solver as the one proposed in this paper, two different class of mesh truncation techniques may be considered. The first class is based on local boundary conditions that are imposed on the external boundary such as ABC [19] or perfectly matched layers (PML) [20]. However, these methods are approximate, even using a higher-order ABC or an optimized PML, and will result in a large computational domain. The second class is based on boundary integral equations (BIE) derived using an appropriate Green's function such as the so-called finite element boundary integral (FE-BI) [21] method. Although, this method is exact and can reduce the size of the computational domain, the final system of equations leads to a partly full and partly sparse matrix with the consequent drawback when solving that system. To date, there is no efficient direct solver developed specifically for solving this type of matrix equation. In addition, the implementation of this method combined with DDM in large-scale parallel is relatively complicated since we must consider both the FEM-DDM and the integral equation (IE)-DDM systems.

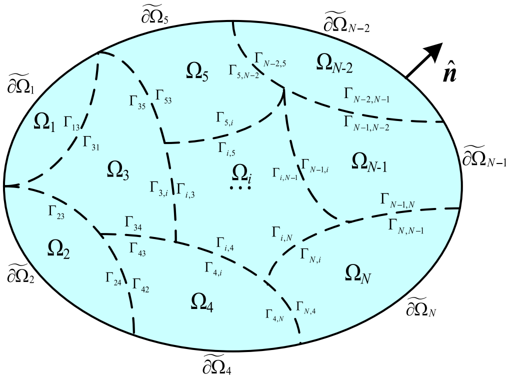


FIGURE 1. Typical DDM setup where the original computational domain is divided into *N* non-overlapping subdomains.

In the particular case of the proposed FEM solver, a (numerically) exact mesh truncation technique, called finite element-iterative integral equation evaluation (FE-IIEE) is implemented and combined with the DDM technology in order to provide very accurate and efficient solutions to challenging scattering and antenna problems. The FE-IIEE technique is based on a non-standard multiplicative Schwarz iterative domain decomposition paradigm between the interior and exterior problems [22]–[24]. The truncation boundary in FE-IIEE may be arbitrarily shaped and, typically, may be placed very close to the objects (in the range of 0.05λ to 0.2λ), reducing the number of unknowns, and consequently, the computational resources. Finally, in order to accelerate the scattering field calculation process required by this truncation method, a parallel efficient version of the multilevel fast multipole algorithm (MLFMA) is used [25].

The rest of this paper is organized as follows. The mathematical approach, including the variation formulation, the DDM implementation and details about the exact mesh truncation technology, is introduced in section II. In section III, a scalable parallel scheme of DDM for supercomputers is presented. In the numerical results section, the performance and accuracy of the proposed solver are validated comparing the results with those given by commercial software such as HFSS and FEKO. Also, the parallel efficiency over ten thousand CPU cores is reported. Finally, some conclusions are gathered in section V.

II. MATHEMATICAL APPROACH

A. VARIATIONAL FORMULATION

Let us start considering a typical DDM setup where the original computational domain is divided into N non-overlapping subdomains as illustrated in Fig. 1. The interfaces between the subdomains are denoted as $\Gamma_{ij} \equiv \partial\Omega_i \cap \partial\Omega_j$ and the exterior boundaries are denoted as $\widetilde{\partial\Omega}_i = \partial\Omega_i \cap \partial\Omega$. Obviously

$\Gamma_{ij} = \Gamma_{ji}$, and $\Gamma_{ij} = \Gamma_{ji} = \emptyset$ when the subdomain Ω_i and Ω_j are not adjacent. Note that Γ_{ij} is used when Ω_i is the computational domain and, on the other hand, Γ_{ji} is used when Ω_j is computed.

In the absence of nonlinear materials, the boundary value problem (BVP) for any subdomain Ω_i in terms of the electric field ($\boldsymbol{E}$) can be expressed as

$$\nabla \times \mu_{ri}^{-1} \nabla \times \boldsymbol{E}_i - k_0^2 \varepsilon_{ri} \boldsymbol{E}_i = -jk_0 \eta_0 \boldsymbol{J}_i^{imp} \text{in } \Omega_i \tag{1}$$

$$\widehat{\boldsymbol{n}}_i \times \boldsymbol{E}_i = 0 \quad \text{on } \Gamma_{\text{PEC}}^i \tag{2}$$

$$\widehat{\boldsymbol{n}}_i \times \left(\mu_{ri}^{-1} \nabla \times \boldsymbol{E}_i\right) = 0 \quad \text{on } \Gamma_{\text{PMC}}^i \tag{3}$$

$$\widehat{\boldsymbol{n}}_i \times \left(\mu_{ri}^{-1} \nabla \times \boldsymbol{E}_i\right) - jk_0 \widehat{\boldsymbol{n}}_i \times \left(\boldsymbol{E}_i \times \widehat{\boldsymbol{n}}_i\right) = \boldsymbol{\Psi} \quad \text{on } \widetilde{\partial \Omega}_i \tag{4}$$

$$\begin{aligned}&\widehat{\boldsymbol{n}}_i \times \left(\mu_{ri}^{-1} \nabla \times \boldsymbol{E}_i\right) + jk_0 \widehat{\boldsymbol{n}}_i \times \left(\boldsymbol{E}_i \times \widehat{\boldsymbol{n}}_i\right) \\ &\quad = -\widehat{\boldsymbol{n}}_j \times \left(\mu_{rj}^{-1} \nabla \times \boldsymbol{E}_j\right) + jk_0 \widehat{\boldsymbol{n}}_j \times \left(\boldsymbol{E}_j \times \widehat{\boldsymbol{n}}_j\right) \quad \text{on } \Gamma_{ij}\end{aligned} \tag{5}$$

where $\widehat{\boldsymbol{n}}_i$ denotes the outward-directed unit normal vector to Γ_{ij}, μ_{ri} and ε_{ri} are the relative permeability and permittivity of the material, $k_0$is the wave number in free space, η_0 is the free space wave impedance, $\boldsymbol{J}_i^{imp}$ is the given impressed current density in Ω_i and j represents the imaginary unit. Equations (2) and (3) represent the Dirichlet and Neumann boundary conditions, respectively. Equation (4) represents the first order ABC imposed on the truncation boundary $\widetilde{\partial \Omega}_i$ where the initial value of $\boldsymbol{\Psi}$ is the result of evaluating (4) with $\boldsymbol{E}\ \mathcal{D}\ \boldsymbol{E}^{inc}$ being $\boldsymbol{E}^{inc}$ the incident electric field over the boundary. The value of $\boldsymbol{\Psi}$ when analyzing an antenna (radiation problem) is zero. As mentioned in the introduction, the proposed solver truncates the exterior infinite domain by an integral equation representation of the exterior field. Then, the solution is obtained through an iterative process in which the residual of the radiation boundary condition on the mesh truncation boundary $\boldsymbol{\Psi}$ is updated. More details about the implementation of this truncation technique are given in Section II. C. Equation (5) is the first order Robin-type transmission condition [5] adopted to enforce the continuity of tangential electric field and magnetic field on the interface.

In order to simplify the previous equations, a new set of auxiliary variables or "cement" surface variable may be defined as in [6]. This "cement" variables are given by

$$\boldsymbol{j}_i = \frac{1}{k_0} \widehat{\boldsymbol{n}}_i \times \left(\mu_{ri}^{-1} \nabla \times \boldsymbol{E}_i\right) \tag{6}$$

$$\boldsymbol{e}_i = \widehat{\boldsymbol{n}}_i \times \left(\boldsymbol{E}_i \times \widehat{\boldsymbol{n}}_i\right) = \boldsymbol{E}_i^t \tag{7}$$

where $\boldsymbol{e}_i$ and $\boldsymbol{j}_i$ are tangential electric fields and surface electric currents, respectively. With the aid of (6) and (7), the transmission condition expressed in (5) can then be written as

$$-jk_0 \boldsymbol{j}_i + k_0 \boldsymbol{E}_i^t = jk_0 \boldsymbol{j}_j + k_0 \boldsymbol{E}_j^t \tag{8}$$

Thus, the variational formulation of the problem described in (1)-(4) and (8) is: find $\boldsymbol{E}_i \in \boldsymbol{W}_i$ and $\boldsymbol{j}_i \in \boldsymbol{V}_i$ such that

$$a\left(\boldsymbol{F}_i, \boldsymbol{E}_i\right) + t\left(\boldsymbol{F}_i, \boldsymbol{j}_i\right) = l\left(\boldsymbol{F}_i\right), \quad \forall \boldsymbol{F}_i \in \boldsymbol{W}_i \tag{9}$$

and

$$c\left(\boldsymbol{F}_i, \boldsymbol{j}_i\right) + t\left(\boldsymbol{e}_i, \boldsymbol{F}_i\right)^T = -c\left(\boldsymbol{F}_i, \boldsymbol{j}_j\right) + t\left(\boldsymbol{e}_j, \boldsymbol{F}_i\right)^T, \quad \forall \boldsymbol{F}_i \in \boldsymbol{W}_i \tag{10}$$

where the bilinear and linear forms, $a\left(\boldsymbol{F}_i, \boldsymbol{E}_i\right)$, $t\left(\boldsymbol{F}_i, \boldsymbol{j}_i\right)$, $c\left(\boldsymbol{F}_i, \boldsymbol{j}_i\right)$ and $l\left(\boldsymbol{F}_i\right)$ are defined as follows

$$\begin{aligned}a\left(\boldsymbol{F}_i, \boldsymbol{E}_i\right) = &\iiint_{\Omega_i} \nabla \times \boldsymbol{F}_i \cdot \mu_{ri}^{-1} \nabla \times \boldsymbol{E}_i \boldsymbol{d}v \\ &- k_0^2 \varepsilon_{ri} \iiint_{\Omega_i} \boldsymbol{F}_i \cdot \boldsymbol{E}_i \boldsymbol{d}v \\ &+ jk_0 \iint_{\widetilde{\partial \Omega}_i} \left(\widehat{\boldsymbol{n}}_i \times \boldsymbol{F}_i\right)\left(\widehat{\boldsymbol{n}}_i \times \boldsymbol{E}_i\right) ds\end{aligned} \tag{11}$$

$$t\left(\boldsymbol{F}_i, \boldsymbol{j}_i\right) = k_0 \iint_{\Gamma_{ij}} \boldsymbol{F}_i \cdot \boldsymbol{j}_i ds \tag{12}$$

$$c\left(\boldsymbol{F}_i, \boldsymbol{j}_i\right) = -jk_0 \iint_{\Gamma_{ij}} \boldsymbol{F}_i \cdot \boldsymbol{j}_i ds \tag{13}$$

$$l\left(\boldsymbol{F}_i\right) = -jk_0 \eta_0 \iiint_{\Omega_i} \boldsymbol{F}_i \cdot \boldsymbol{J}_i^{imp} \mathrm{d}v - \iint_{\widetilde{\partial \Omega}_i} \boldsymbol{F}_i \cdot \Psi_i ds \tag{14}$$

with

$$\boldsymbol{W}_i := \left\{\boldsymbol{A}_i \in \boldsymbol{H}\left(curl, \Omega\right), \widehat{\boldsymbol{n}} \times \boldsymbol{A}_i = 0 \text{ on } \Gamma_{\text{PEC}}\right\} \tag{15}$$

where $\boldsymbol{H}\left(curl, \Omega\right)$ is the space of square integrable vector functions with square integrable curl, and

$$\boldsymbol{V}_i := \left\{\boldsymbol{B}_i \in \boldsymbol{H}^{-1/2}\left(curl_\tau, \Gamma_{ij}\right)\right\} \tag{16}$$

According to (9) and (10), the final FEM system can be transformed into a matrix equation form as shown in (17), as shown at the top of the next page, where the superscript $\boldsymbol{I}$ refers to the unknown coefficients inside each subdomain and the superscripts $\boldsymbol{e}$ and $\boldsymbol{j}$ are the tangential electric field and current unknown coefficients on interface Γ_{ij}, respectively. The subscript of the matrix denotes the index of subdomain.

Further, the matrix equation (17) can be written in a compact form as

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{C}_{12} & \cdots & \boldsymbol{C}_{1N} \\ \boldsymbol{C}_{21} & \boldsymbol{A}_2 & \cdots & \boldsymbol{C}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{N1} & \boldsymbol{C}_{N2} & \cdots & \boldsymbol{A}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_N \end{bmatrix} \tag{18}$$

where the matrix $\boldsymbol{A}_i$ is the self-region submatrix, the matrix $\boldsymbol{C}_{ij}$ is the coupling matrix between subdomain Ω_i and subdomain Ω_j, $\boldsymbol{b}_i$ is the right hand side excitation with $\boldsymbol{b}_i =$

$$\begin{bmatrix}
a\left(\boldsymbol{F}_1^{\boldsymbol{I}}, \boldsymbol{E}_1^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_1^{\boldsymbol{I}}, \boldsymbol{E}_1^{\boldsymbol{e}}\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
a\left(\boldsymbol{F}_1^{\boldsymbol{e}}, \boldsymbol{E}_1^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_1^{\boldsymbol{e}}, \boldsymbol{E}_1^{\boldsymbol{e}}\right) & t\left(\boldsymbol{F}_1^{\boldsymbol{e}}, \boldsymbol{j}_1\right) & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & t\left(\boldsymbol{F}_1^{\boldsymbol{e}}, \boldsymbol{j}_1\right)^T & c\left(\boldsymbol{F}_1^{\boldsymbol{j}}, \boldsymbol{j}_1\right) & \mathbf{0} & -t\left(\boldsymbol{F}_1^{\boldsymbol{e}}, \boldsymbol{j}_2\right)^T & c\left(\boldsymbol{F}_1^{\boldsymbol{j}}, \boldsymbol{j}_2\right) & \cdots & \mathbf{0} & -t\left(\boldsymbol{F}_1^{\boldsymbol{e}}, \boldsymbol{j}_N\right)^T & c\left(\boldsymbol{F}_1^{\boldsymbol{j}}, \boldsymbol{j}_N\right) \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & a\left(\boldsymbol{F}_2^{\boldsymbol{I}}, \boldsymbol{E}_2^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_2^{\boldsymbol{I}}, \boldsymbol{E}_2^{\boldsymbol{e}}\right) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & a\left(\boldsymbol{F}_2^{\boldsymbol{e}}, \boldsymbol{E}_2^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_2^{\boldsymbol{e}}, \boldsymbol{E}_2^{\boldsymbol{e}}\right) & t\left(\boldsymbol{F}_2^{\boldsymbol{e}}, \boldsymbol{j}_2\right) & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & -t\left(\boldsymbol{F}_2^{\boldsymbol{e}}, \boldsymbol{j}_1\right)^T & c\left(\boldsymbol{F}_2^{\boldsymbol{j}}, \boldsymbol{j}_1\right) & \mathbf{0} & t\left(\boldsymbol{F}_2^{\boldsymbol{e}}, \boldsymbol{j}_2\right)^T & c\left(\boldsymbol{F}_2^{\boldsymbol{j}}, \boldsymbol{j}_2\right) & \cdots & \mathbf{0} & -t\left(\boldsymbol{F}_2^{\boldsymbol{e}}, \boldsymbol{j}_N\right)^T & c\left(\boldsymbol{F}_2^{\boldsymbol{j}}, \boldsymbol{j}_N\right) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & a\left(\boldsymbol{F}_N^{\boldsymbol{I}}, \boldsymbol{E}_N^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_N^{\boldsymbol{I}}, \boldsymbol{E}_N^{\boldsymbol{e}}\right) & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & a\left(\boldsymbol{F}_N^{\boldsymbol{e}}, \boldsymbol{E}_N^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_N^{\boldsymbol{e}}, \boldsymbol{E}_N^{\boldsymbol{e}}\right) & t\left(\boldsymbol{F}_N^{\boldsymbol{e}}, \boldsymbol{j}_N\right) \\
\mathbf{0} & -t\left(\boldsymbol{F}_N^{\boldsymbol{e}}, \boldsymbol{j}_1\right)^T & c\left(\boldsymbol{F}_N^{\boldsymbol{j}}, \boldsymbol{j}_1\right) & \mathbf{0} & -t\left(\boldsymbol{F}_N^{\boldsymbol{e}}, \boldsymbol{j}_2\right)^T & c\left(\boldsymbol{F}_N^{\boldsymbol{j}}, \boldsymbol{j}_2\right) & \cdots & \mathbf{0} & t\left(\boldsymbol{F}_N^{\boldsymbol{e}}, \boldsymbol{j}_N\right)^T & c\left(\boldsymbol{F}_N^{\boldsymbol{j}}, \boldsymbol{j}_N\right)
\end{bmatrix}
\times \begin{bmatrix} \boldsymbol{x}_1^{\boldsymbol{I}} \\ \boldsymbol{x}_1^{\boldsymbol{e}} \\ \boldsymbol{x}_1^{\boldsymbol{j}} \\ \boldsymbol{x}_2^{\boldsymbol{I}} \\ \boldsymbol{x}_2^{\boldsymbol{e}} \\ \boldsymbol{x}_2^{\boldsymbol{j}} \\ \vdots \\ \boldsymbol{x}_N^{\boldsymbol{I}} \\ \boldsymbol{x}_N^{\boldsymbol{e}} \\ \boldsymbol{x}_N^{\boldsymbol{j}} \end{bmatrix}
= \begin{bmatrix} l\left(\boldsymbol{F}_1^{\boldsymbol{I}}\right) \\ l\left(\boldsymbol{F}_1^{\boldsymbol{e}}\right) \\ \mathbf{0} \\ l\left(\boldsymbol{F}_2^{\boldsymbol{I}}\right) \\ l\left(\boldsymbol{F}_2^{\boldsymbol{e}}\right) \\ \mathbf{0} \\ \vdots \\ l\left(\boldsymbol{F}_N^{\boldsymbol{I}}\right) \\ l\left(\boldsymbol{F}_N^{\boldsymbol{e}}\right) \\ \mathbf{0} \end{bmatrix} \tag{17}$$

$l\left(\boldsymbol{F}_i^{\boldsymbol{X}}\right)$ and $\boldsymbol{x}_i = [\boldsymbol{x}_i^{\boldsymbol{I}}, \boldsymbol{x}_i^{\boldsymbol{e}}, \boldsymbol{x}_i^{\boldsymbol{j}}]^T$ is the unknown coefficients of Ω_i. Note that, if subdomain Ω_i and Ω_j are not adjacent, matrix $\boldsymbol{C}_{ij} = 0$. Then, matrix $\boldsymbol{A}_i$ may be expressed as

$$\boldsymbol{A}_i = \begin{bmatrix} a\left(\boldsymbol{F}_i^{\boldsymbol{I}}, \boldsymbol{E}_i^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_i^{\boldsymbol{I}}, \boldsymbol{E}_i^{\boldsymbol{e}}\right) & \mathbf{0} \\ a\left(\boldsymbol{F}_i^{\boldsymbol{e}}, \boldsymbol{E}_i^{\boldsymbol{I}}\right) & a\left(\boldsymbol{F}_i^{\boldsymbol{e}}, \boldsymbol{E}_i^{\boldsymbol{e}}\right) & t\left(\boldsymbol{F}_i^{\boldsymbol{e}}, \boldsymbol{j}_i\right) \\ \mathbf{0} & t\left(\boldsymbol{F}_i^{\boldsymbol{e}}, \boldsymbol{j}_i\right)^T & c\left(\boldsymbol{F}_i^{\boldsymbol{j}}, \boldsymbol{j}_i\right) \end{bmatrix} \tag{19}$$

and matrix $\boldsymbol{C}_{ij}$ as

$$\boldsymbol{C}_{ij} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -t\left(\boldsymbol{F}_i^{\boldsymbol{e}}, \boldsymbol{j}_j\right)^T & c\left(\boldsymbol{F}_i^{\boldsymbol{j}}, \boldsymbol{j}_j\right) \end{bmatrix} \tag{20}$$

The discretization of the above variational formulation is achieved by using our own versions of higher-order isoparametric curl-conforming basis functions that constitute a rigorous implementation of Nédélec first family of finite elements [26], [27]. These functions are obtained by systematic approach based on the *a priori* definition on a reference element of the space of the basis functions and the obtainment of the basis functions as the dual basis with respect to a set of unisolvent degrees of freedom acting on the defined space. It is worth noting that the basis functions are obtained in the reference finite element and are transformed to the real one using the inverse of the Jacobian matrix. Details about the methodology used to get the solution for the above variational formulation are given next.

B. SOLUTION METHODOLOGY

Once the variational formulation of the problem is obtained, the next step is to get the solution to that system of equations. One may think that the easier way to solve (17) is to use a direct solver and solve the whole matrix equation directly. However, this is not a desirable choice since the size of the matrix may make the problem unapproachable, computationally speaking. On the other hand, the inverse of each self-region submatrix $\boldsymbol{A}_i$ may be used as preconditioner in order to obtain an easier to solve system of equation. Taking (18) and multiplying each term by the inverse of the corresponding self-region submatrix we have

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}_1^{-1}\boldsymbol{C}_{12} & \cdots & \boldsymbol{A}_1^{-1}\boldsymbol{C}_{1N} \\ \boldsymbol{A}_2^{-1}\boldsymbol{C}_{21} & \boldsymbol{I} & \cdots & \boldsymbol{A}_2^{-1}\boldsymbol{C}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_N^{-1}\boldsymbol{C}_{N1} & \boldsymbol{A}_N^{-1}\boldsymbol{C}_{N2} & \cdots & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1^{-1}\boldsymbol{b}_1 \\ \boldsymbol{A}_2^{-1}\boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{A}_N^{-1}\boldsymbol{b}_N \end{bmatrix} \tag{21}$$

where $\boldsymbol{I}$ is the identity matrix. It is important to remark that the matrix multiplication $\boldsymbol{A}_i^{-1}\boldsymbol{C}_{ij}$ gives values different from zero only in the interface variables of each subdomain. Thus, the global system of equation in (21) may be expressed only in terms of the interface variables as

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}_1^{-1}\boldsymbol{C}_{12} & \cdots & \boldsymbol{A}_1^{-1}\boldsymbol{C}_{1N} \\ \boldsymbol{A}_2^{-1}\boldsymbol{C}_{21} & \boldsymbol{I} & \cdots & \boldsymbol{A}_2^{-1}\boldsymbol{C}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_N^{-1}\boldsymbol{C}_{N1} & \boldsymbol{A}_N^{-1}\boldsymbol{C}_{N2} & \cdots & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_1 \\ \tilde{\boldsymbol{x}}_2 \\ \vdots \\ \tilde{\boldsymbol{x}}_N \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{b}}_1 \\ \tilde{\boldsymbol{b}}_2 \\ \vdots \\ \tilde{\boldsymbol{b}}_N \end{bmatrix} \tag{22}$$

where $\tilde{\boldsymbol{x}}_i = [\boldsymbol{x}_i^e, \boldsymbol{x}_i^j]^T$, $\tilde{\boldsymbol{b}}_i$ is a reduced right hand side excitation only related to the interface variables.

Then, (22) can be solved using a Krylov subspace iterative method. In particular, the restarted generalized minimal residual (GMRES) method [28] is adopted in this paper. Once the interface unknowns $\tilde{\boldsymbol{x}}_i$, $i = 1, 2, \ldots, N$ are iteratively computed by GMRES, the unknown coefficients inside each subdomain can be independently solved via

$$\boldsymbol{x}_i = \boldsymbol{A}_i^{-1}\left(\boldsymbol{b}_i - \sum_{j \neq i} \boldsymbol{C}_{ij}\tilde{\boldsymbol{x}}_j\right) \tag{23}$$

It is clear that only the self-region submatrix $\boldsymbol{A}_i$ of each domain needs be factorized to compute the problem solution. The present implementation of the FEM solver supports the use of different direct sparse solvers such as MUMPS, MKL PARDISO or SUPERLU [29] to perform this factorization. Thanks to the moderate and controllable size of the self-region submatrix $\boldsymbol{A}_i$, these direct sparse solvers does not suffer from stability problems as happen with the traditional FEM scheme when the size of the problem increases considerably.

C. MESH TRUNCATION TECHNIQUE

As mentioned in the Introduction, an exact mesh truncation technique, called finite element-iterative integral equation evaluation (FE-IIEE), is implemented and combined with the DDM technology to provide very accurate solutions to nowadays challenging scattering and antenna problems. The FE-IIEE methodology is based on a non-standard multiplicative Schwarz iterative domain decomposition paradigm between the interior and exterior problems and it divides the original infinite domain into two overlapping domains: a FEM domain (Ω^{FEM}) bounded by the surface S and the infinite domain exterior to the auxiliary boundary S' (Ω^{EXT}) [22]–[24]. Thus, the overlapping region is limited by S' and S. Figure 2 illustrates a typical setup of the FE-IIEE technique where one can see the boundaries S' and S delimiting the mentioned overlapping region. The exterior infinite domain is truncated by an integral equation representation of the exterior field to S'. Then, the solution is obtained through an iterative process in which the residual of the radiation boundary condition on the mesh truncation boundary ($\boldsymbol{\Psi}$ term in eq. (4)) is updated. As in any iterative algorithm, the convergence is a key issue to consider. In this particular case, the convergence is assured by using convex exterior boundaries being the rate of convergence faster when the overlapping between the interior and exterior domains is larger [22], [30], [31].

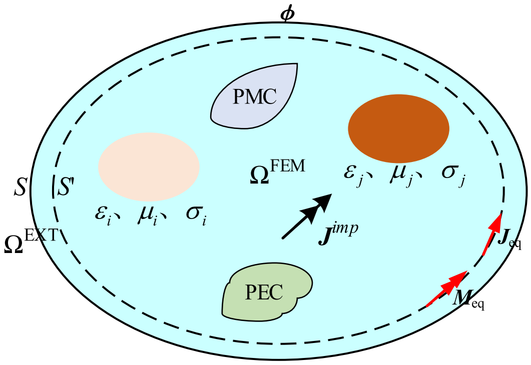

ϕ
PMC
S
S'
Ω^{FEM}
ε_j、 μ_j、 σ_j
ε_i、 μ_i、 σ_i
Ω^{EXT}
$\boldsymbol{J}^{imp}$
$\boldsymbol{J}_{\text{eq}}$
PEC
$\boldsymbol{M}_{\text{eq}}$

FIGURE 2. FE-IIEE method for general electromagnetic field problem.

It is important to remark that the calculation of the integral equation representation of the exterior field to S' may be extremely expensive, computationally speaking, for large challenging electromagnetic problems. Although fully parallelizable, the convolutional character (double loop) of this calculation leads to a computational complexity of O(M^2), being M the number of unknowns associated to the exterior boundary S. For this reason, the present implementation of the FE-IIEE technique makes use of a parallel MLFMA algorithm to accelerate this integral equation calculation [25]. Another important feature of this implementation is the transparent combination with the DDM methodology previously introduced. Thus, the proposed FEM solver is able to perform very accurate analysis for both antenna and scattering problems when the external mesh truncation boundary is very close to the electromagnetic sources (even with distances smaller than 0.05 λ).

The truncation method starts performing an initial FEM analysis of the problem using the Cauchy (Robin) boundary condition expressed in (4) over the external surfaces. Figure 3 shows the flowchart of the present implementation of the FE-IIEE method where this initial FEM analysis is marked using blue squares. Once eq. (17) is solved and the values for the unknown coefficients of each subdomain are obtained, the equivalent currents $\boldsymbol{j}_{eq}$ and $\boldsymbol{M}_{eq}$ over the interior boundary S' are calculated. Then, the scattering field, and its curl, over S radiated by the equivalent currents are computed using MLFMA. The fields radiated by the FEM region, $\boldsymbol{E}_{IIEE}$ and their curl ($\nabla \times \boldsymbol{E}_{IIEE}$), are compute using the following integral expressions

$$\begin{aligned} \boldsymbol{E} = -\frac{k^2}{(4\pi)^2}[\eta \iint_{4\pi} e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{r}_m)} T_L(kR_{m'm}, \hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{R}}_{m'm}) \\ \times (\bar{\boldsymbol{I}} - \hat{k\hat{k}})\widetilde{\boldsymbol{J}}_{m'}(\hat{\boldsymbol{k}})\, d^2\hat{\boldsymbol{k}} - \iint_{4\pi} \widehat{\boldsymbol{k}} \\ \times e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{r}_m)} T_L(kR_{m'm}, \hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{R}}_{m'm})\widetilde{\boldsymbol{M}}_{m'}(\hat{\boldsymbol{k}}) d^2\hat{\boldsymbol{k}}] \end{aligned} \tag{24}$$

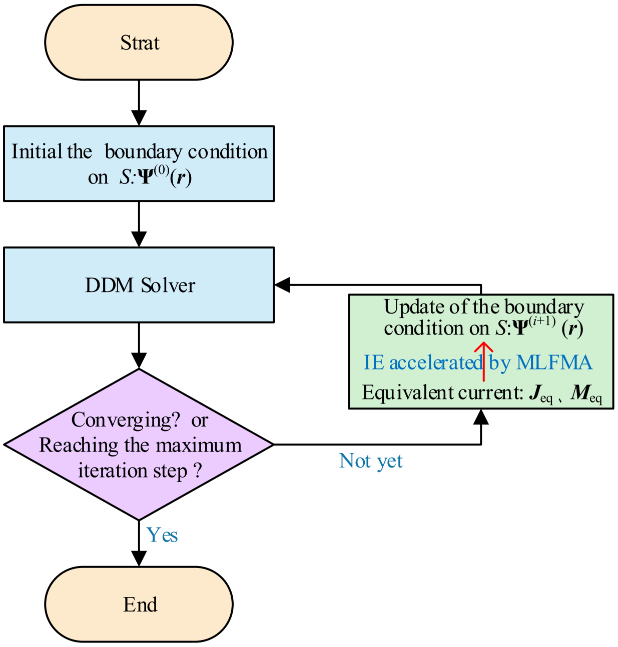

Strat
Initial the boundary condition on S:$\Psi^{(0)}(r)$
DDM Solver
Update of the boundary condition on S:$\Psi^{(i+1)}(r)$
IE accelerated by MLFMA
Equivalent current: J_{eq}、M_{eq}
Converging? or Reaching the maximum iteration step ?
Not yet
Yes
End

FIGURE 3. Basic flowchart of FE-IIEE combined with DDM.

$$\nabla \times \boldsymbol{E} = \frac{jk^3}{(4\pi)^2}[\eta \iint_{4\pi} \hat{\boldsymbol{k}} \times e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{r}_m)} T_L(kR_{m'm}, \hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{R}}_{m'm}) \times \tilde{\boldsymbol{J}}_{m'}(\hat{\boldsymbol{k}})\, d^2\hat{\boldsymbol{k}} + \iint_{4\pi} e^{-j\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{r}_m)} T_L(kR_{m'm}, \hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{R}}_{m'm}) \times (\bar{\boldsymbol{I}} - \hat{k}\hat{k})\tilde{\boldsymbol{M}}_{m'}(\hat{\boldsymbol{k}}) d^2\hat{\boldsymbol{k}}] \tag{25}$$

where $\hat{\boldsymbol{k}}$ is the unit vector of plane wave expansion direction, $\boldsymbol{r}$ is the observed field point coordinates, $\boldsymbol{r}_m$ is the center of filed point group, $\tilde{\boldsymbol{J}}_{m'}(\hat{\boldsymbol{k}})$ and $\tilde{\boldsymbol{M}}_{m'}(\hat{\boldsymbol{k}})$ are the radiation pattern of the electric current and magnetic current, respectively, $\boldsymbol{r}'$ is the source point coordinates, $\boldsymbol{r}'_m$ is the center of source point group, and $T_L(kR_{m'm}, \hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{R}}_{m'm})$ is the translator operator between source point group m' and field point group m. These calculations are illustrated in Fig. 3 using the green square. It is worth pointing out that the MLFMA employed to accelerate the scattering field computation is parallelized using an adaptive direction partitioning strategy with improved load balance which developed in our previous works [32].

Once the scattering fields are obtained, a new value of $\boldsymbol{\Psi}$($\boldsymbol{\Psi}^{i+1}$ in general) is computed by introducing the values of the fields $\boldsymbol{E}_{IIEE}$ and ($\nabla \times \boldsymbol{E}_{IIEE}$) in eq. (4). Finally, the error between iterations, ψ, is calculated. If the error is greater than an error threshold, the method will start again using the new residual function $\boldsymbol{\Psi}^{i+1}$, otherwise the iteration process finishes (step marked in Fig. 3 using the purple rhombus). The error in $\boldsymbol{\Psi}$ is measured in a weighted L$_2$-norm

$$\psi = \frac{\left\| \boldsymbol{\Psi}^{i+1} - \boldsymbol{\Psi}^{i} \right\|_2}{\left\| \boldsymbol{\Psi}^{i+1} \right\|_2} \tag{26}$$

Thus, a (numerically) exact radiation boundary condition is imposed cancelling undesired reflections that, sometimes, come back to the computational domain. This numerical technique also retains the sparsity of the FEM matrices with the consequent improvement in stability during the solving process. It is worth noting that the numerical cost of the second and subsequent iterations is very small since the previous solution is used as starting point in the GMRES algorithm. However, it is worth mentioning that the error between iterations, ψ, is limited by the tolerance of the GMRES algorithm.

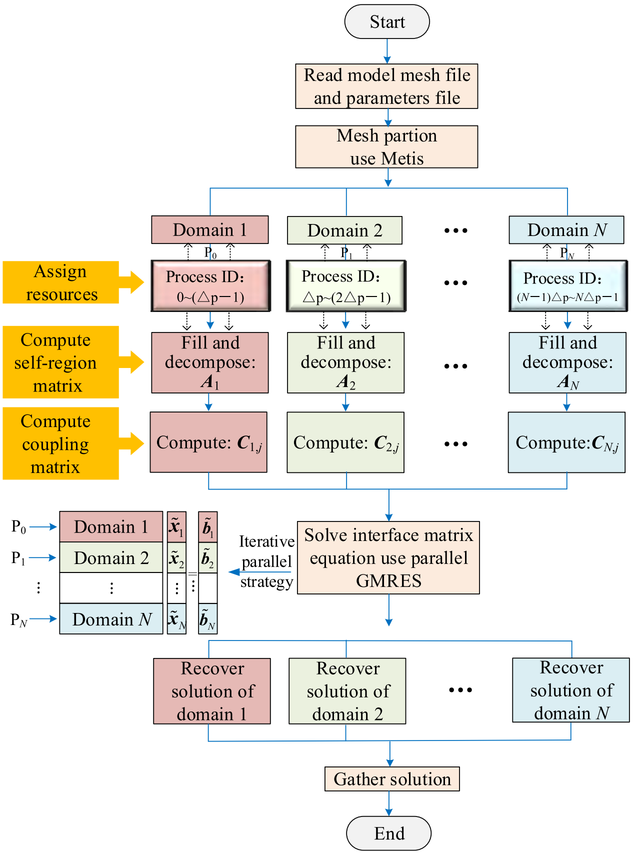

Start
Read model mesh file and parameters file
Mesh partion use Metis
Domain 1
Domain 2
Domain N
P_0
P_1
P_N
Assign resources
Process ID: 0~(△p−1)
Process ID: △p~(2△p−1)
Process ID: (N−1)△p~N△p−1
Compute self-region matrix
Fill and decompose: A_1
Fill and decompose: A_2
Fill and decompose: A_N
Compute coupling matrix
Compute: $C_{1,j}$
Compute: $C_{2,j}$
Compute: $C_{N,j}$
$\tilde{x}_1$
$\tilde{x}_2$
$\tilde{x}_N$
$\tilde{b}_1$
$\tilde{b}_2$
$\tilde{b}_N$
Iterative parallel strategy
Solve interface matrix equation use parallel GMRES
Recover solution of domain 1
Recover solution of domain 2
Recover solution of domain N
Gather solution
End

FIGURE 4. Parallel framework for the solver.

III. MASSIVELY PARALLEL IMPLEMENTATION

One of the key features of the proposed FEM solver is its massively parallel implementation. Although the DDM technology is inherently parallelizable, achieving high parallel efficiency and stability when the computing platform scales over thousands CPU cores is still a challenge. In general, the load balancing and the Computation to Communication Ratio (CCR) are two major factors that affect the parallel performance of a program. Thus, how to balance the load, improve the CCR as much as possible and ultimately form a flexible parallel strategy for the solver are the focus of this section.

For a clear description, the parallel flowchart of the FEM solver solution process is sketched in Fig. 4. Obviously, after reading in the necessary model mesh information and parameters file, both data and parallel tasks are allocated in each processor considering the load balance effects of the

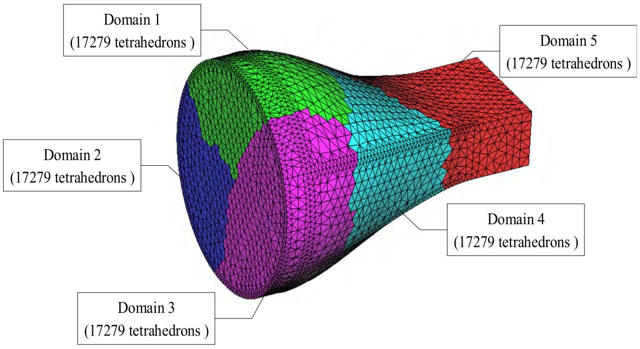
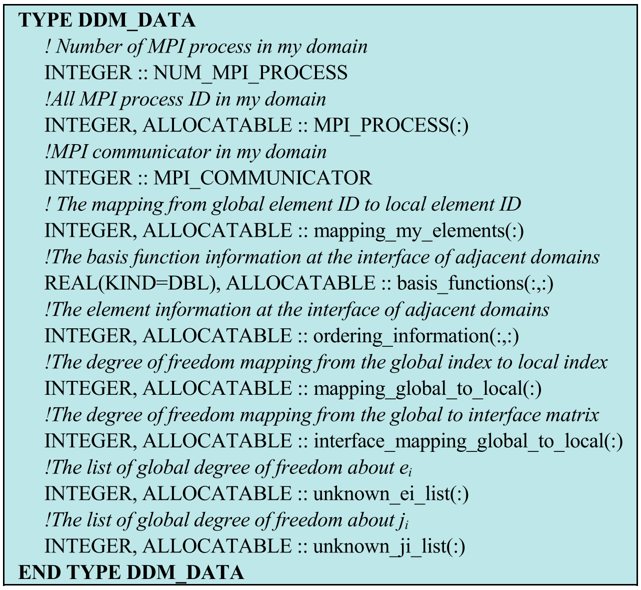


FIGURE 5. The decompositions of a circular horn antenna's global mesh into five subdomains.



```
TYPE DDM_DATA
    ! Number of MPI process in my domain
    INTEGER :: NUM_MPI_PROCESS
    !All MPI process ID in my domain
    INTEGER, ALLOCATABLE :: MPI_PROCESS(:)
    !MPI communicator in my domain
    INTEGER :: MPI_COMMUNICATOR
    ! The mapping from global element ID to local element ID
    INTEGER, ALLOCATABLE :: mapping_my_elements(:)
    !The basis function information at the interface of adjacent domains
    REAL(KIND=DBL), ALLOCATABLE :: basis_functions(:,:)
    !The element information at the interface of adjacent domains
    INTEGER, ALLOCATABLE :: ordering_information(:,:)
    !The degree of freedom mapping from the global index to local index
    INTEGER, ALLOCATABLE :: mapping_global_to_local(:)
    !The degree of freedom mapping from the global to interface matrix
    INTEGER, ALLOCATABLE :: interface_mapping_global_to_local(:)
    !The list of global degree of freedom about e_i
    INTEGER, ALLOCATABLE :: unknown_ei_list(:)
    !The list of global degree of freedom about j_i
    INTEGER, ALLOCATABLE :: unknown_ji_list(:)
END TYPE DDM_DATA
```

FIGURE 6. The date structure designed for the subdomain information.

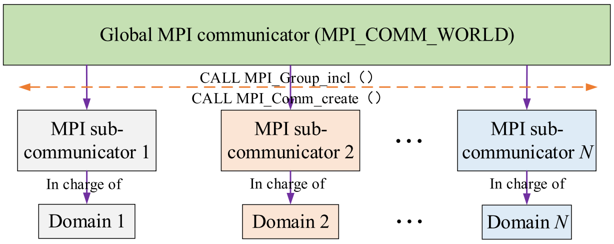


FIGURE 7. Creating the MPI sub-communicator for each subdomain.

whole computing platform. The amount of data and tasks allocated on these processors are closely tied to ensure an efficient load balance. Thus, the solver may use all the computing power and storage capacity of large high-performance computing (HPC) clusters or supercomputers. An automatic parallel domain partition algorithm as ParMETIS [33] is used to distribute the mesh along the processor. In detail, the code starts dividing the global FEM mesh into several subdomains using the mentioned ParMETIS package. As an example, Figure 5 shows the decomposition of a circular horn antenna into five subdomains where the number of tetrahedrons that belongs to each subdomain is completely the same leading to a similar amount of work on them. These characteristics help us to minimize the total communication volume and distributed the load uniformly between the compute nodes. In this load distribution methodology, we have

$$T_1 \approx T_2 \approx \cdots \approx T_N \tag{27}$$

here T_i is the calculation task of the i-th subdomain.

Once the global mesh is decomposed, a certain number of computing resources should be assigned to each subdomain. Clearly, for a good load balance strategy, the following equation must be satisfied

$$\frac{T_1}{F_1} \approx \frac{T_2}{F_2} \approx \cdots \approx \frac{T_N}{F_N} \tag{28}$$

where F_i is the computing resources assigned to i-th subdomain. According to (27), in order to achieve a balanced load, we should have

$$F_1 \approx F_2 \approx \cdots \approx F_N \tag{29}$$

that indicates that the computing resources assigned to each subdomain must be the same.

In order to obtain a highly efficient parallel performance and provide a better adaption with the current supercomputer architectures, the hybrid parallel paradigm that combines a distributed-memory scheme (MPI) with a shared-memory scheme (OpenMP) is adopted. The MPI programming model is used to implement scalability among different compute nodes. The latter is aimed to accelerate computation in the same compute node and relieve the storage pressure of MPI process. At the same time, different combinations of MPI process plus OMP threads can be used to optimize the solver depending on the configuration of each computing platform. Due to (29), the number of MPI process and threads assigned to each subdomain should be identical to ensure load balance (note that we consider uniform computing platforms where all the compute nodes have the same architecture). Figure 4 shows this process under the assign resources label where the Δp number indicates the number of MPI process assigned to the calculation of each subdomain. Simultaneously, every MPI process spawns a same predefined number of threads into the CPU cores of the system.

It is important to point out that there may be three different resource allocations depending on the number of subdomains and the number of processes: one MPI process solves more than one subdomain; one MPI process solves one subdomain; and many MPI processes solve one subdomain simultaneously. In order to make the program effectively parallel in any situation, it is necessary to design a suitable data structure that contains the number of processes, the MPI process ID, the MPI communicator, the local mesh information and so on for each subdomain. The data structure designed in this paper is shown in Fig. 6. The meaning of each variable in the data structure is explained by the italic body in the figure. Specially, since we use the peer-to-peer mode to carry out the parallelization of the program, the MPI sub-communicators responsible for each subdomain should be created using the process shown in Fig. 7, when many MPI processes solve one subdomain concurrently. The sub-communicator information is stored in the integer variable ddm_data%mpi_communicator.

On the other hand, before continuing with the execution of the program, it is necessary that each domain gathers the required basis function information $\boldsymbol{j}_j$ that defined on the interface of its adjacent subdomains to compute the coupling matrix $\boldsymbol{C}_{ij}$ given in (20). Thus, any undesired communication will be completely avoided later to improve the CCR. These basis function are calculated before the global mesh is freed up and then stored in the real array ddm_data%basis_functions(:,:). This processing will greatly improve the parallel efficiency and stability of the program, especially when tens of thousands CPU cores are used, and the electrically large problem is divided into hundreds of subdomains for calculation. Specifically, assume that the number of degree of freedoms (DOFs) at the interface of all subdomains is ten million for a certain problem. Then, the total amount of data about basis functions $\boldsymbol{j}_j$ is about $6*3*8*10\wedge^7 = 1.34$GB (here, the number 6 is the number of Gaussian points on a triangle, 3 stands by the three directions of x, y, and z, and 8 indicates real double precision). In other words, 1.34GB MPI communications will be avoided in this problem through this treatment, drastically improving the CCR of the code.

After the load distribution step, pre-processing calculations get the necessary subdomain information and each MPI process starts the factorization of the self-domain matrix $\boldsymbol{A}_i$, the calculation of the corresponding coupling matrix $\boldsymbol{C}_{ij}$ and the calculation of the excitation vector $\tilde{\boldsymbol{b}}_i$, independently. If more than one process is in charge of one subdomain, these calculations are performed in parallel between them controlled by the respective sub-communicator. Due to the careful load distribution performed in the previous step, the sizes of the self-domain matrices $\boldsymbol{A}_i$ and the corresponding factorization coefficients are very similar between processes. This implies that, typically, the processes will complete their calculation almost at the same time, ensuring high parallel efficiency since there is no undesired waits. Figure 4 shows this process under the labels computing self-region matrix and computing coupling matrix.

Once the calculation of each subdomain finishes, the interface matrix equation (22) is constructed and, then, solved by GMRES in parallel. As we know from the DDM theory, the unknown coefficients corresponding to the interface between subdomains are independent from one side to the other. Then, in the case of the proposed solver, the unknown coefficients in the interface matrix equation are ordered by subdomains where the first rows contain the unknown coefficients corresponding to the first subdomain, the next rows contain the unknown coefficients corresponding to the second subdomain and so on. Thus, the parallelization of the GMRES algorithm is straightforward using a row-oriented MPI process partitioning strategy as shown in Fig. 4. During the matrix–vector multiplication process $\tilde{\boldsymbol{v}}_i = \left[\boldsymbol{I} + \sum_{j=1,j\neq i}^{N} \boldsymbol{A}_i^{-1}\boldsymbol{C}_{ij}\right]\tilde{\boldsymbol{x}}_j^{(k-1)}$(at kth iteration), the required matrix information and its factorization are already in memory from the previous step. It is worth noting that the information exchange between MPI processes only occurs when getting the vector $\tilde{\boldsymbol{x}}_j^{(k-1)}(j \neq i)$. The $\tilde{\boldsymbol{x}}^{(k-1)}$ is accumulated and redistributed by the global communication using an all reduce operation. Once the solution of the interface problem is obtained, each subdomain is able to solve its self-domain problem independently and concurrently in parallel by using the calculated boundary condition at the subdomain interfaces. Then, the global solution is obtained using a gather operation.

As a summary, the parallelization of the DDM described above provides a small DOFs on subdomain interfaces and a complete balanced load. The communications only involve once ***allreduce*** sum operation in each iteration. Thus, this guarantees high parallel efficiency and stability of the program when it scales up to tens of thousands of CPU cores.

IV. NUMERICAL EXPERIMENTS

In this section, different benchmarks are carried out in order to validate the performance, versatility and the results accuracy of our FEM solver. Comparisons with well-known commercial software such as ANSYS HFSS [34] and FEKO [35] and with already validated in-house codes are done to accomplish this validation task. The MKL PARDISO solver is used to solve the subdomain matrix equation in these simulations. All the benchmark simulations were performed in a large-size HPC cluster equipped with 548 compute nodes if no particular indication is used. Each node has two 32-core AMD HygonGenuine 2.0 GHz processors (32 × 512 KB L2 Cache and 64 MB L3 Cache) and 256 GB RAM. The compute nodes are connected with InfiniBand switches to provide the highest communication speeds.

A. ACCURACY VALIDATION

This first benchmark consists of the electromagnetic analysis of three different models such as a waveguide slot antenna, a modern missile and a low-scattering object. The results of each simulation are compared with those given by the aforementioned commercial softwares. Thus, this benchmark may be considered as an accuracy validation test for our FEM solver.

1) WAVEGUIDE WIDE-EDGE SLOT ANTENNA

The model consists of a waveguide wide-edge slot antenna with 64 slots along the propagation direction. The antenna has two radiating waveguides and one more for feeding purposes. Figure 8 shows the geometry of this waveguide slot antenna. The operation frequency is 18.7 GHz. and a coaxial wave port placed at the middle of the feeding waveguide is

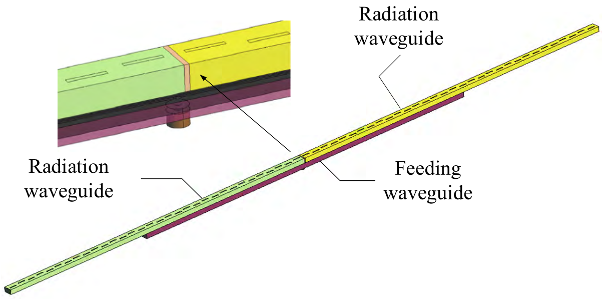


FIGURE 8. **Waveguide wide-edge slot antenna model.**

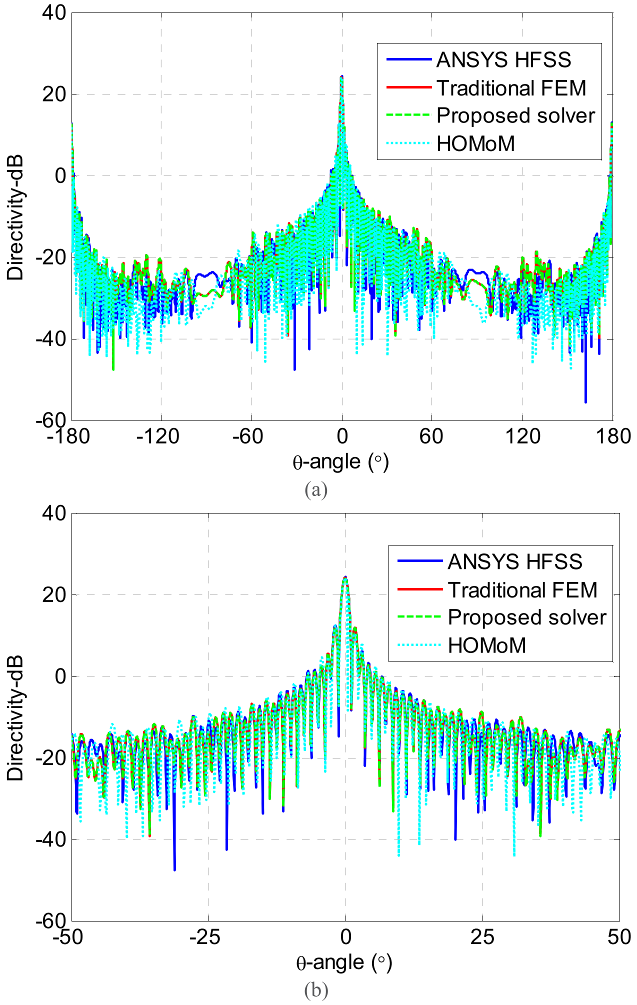
TABLE 1. **Computational statistics of proposed solver and traditional Fem for the waveguide wide-edge slot antenna.**

Methods	CPU cores	Unknowns	Iterations with ξ $=10^{-3}$	Peak Memory Usage/GB	Total CPU time/s
Traditional FEM	128	9,861,370	/	258.5	684.4
Proposed	128	10,492,916	159	99.1	207.1
ANSYS HFSS	12	9,809,307	/	59.0	2580.3
Proposed	12	10,474,476	144	97.3	1044.9

used to excite the antenna. The relative permittivity of the material used in the coaxial cable is set to 2.03. The total number of tetrahedron used in the discretization of the model is 1,503,663 and they are divided into 64 subdomains. The tolerance for the parallel GMRES solver is set to $\xi = 10^{-3}$. It is important to mention that the mesh truncation technique employed in this case is the first order ABC. The use of the FE-IIEE technique will be introduced later, in the third example of this benchmark, once the electromagnetic results given by the solver are validated using the well-known first order ABC.

The simulation results are compared with those given by the traditional FEM version of the solver without DDM, the commercial software HFSS and an in-house higher order method of moments (HOMoM) code [36]. The traditional FEM version uses the MKL PARDISO solver to obtain the solution of the large complex sparse matrix equation directly in parallel. The solver used in HFSS is its default direct solver option. Figure 9 shows the comparison of the radiation pattern for the elevation cut (YOZ plane) where a very good agreement is appreciated.

The computational statistics for this simulation are given in Table 1. The FEM solver implemented in this paper significantly reduces the peak memory usage and reduces the total calculation time compared to the traditional FEM methodology using the same computing resources in an HPC cluster. The platform employed for the ANSYS HFSS simulation and the corresponding FEM simulation using the proposed solver is a workstation equipped with 24 CPU cores and 192 GB of RAM (the model is divided into 60 subdomains for the proposed solver). It is worth noting that the ANSYS HFSS in-core solver exceeds the memory limit (192 GB), so the out-of-core solver is adopted. In addition to the RAM, the out-of-core solver uses the space on the hard disk whereas the proposed solver uses only the RAM, so the behavior, as listed in Table 1, that the peak memory with the proposed method is larger than the one with ANSYS HFSS solver is completely reasonable. Thus, compared with HFSS, the proposed solver is more efficient when solving a same size problem using the same computing resources.



(a)

(b)

FIGURE 9. **Radiation pattern (YOZ plane) of the waveguide slot antenna model :(a) Angle from −180° to 180°. (b) Angle from −50° to 50°.**

2) MODERN MISSILE

The second simulation consists of the bistatic radar cross-section (RCS) analysis of a PEC modern missile as the one shown in Fig. 10. The working frequency is 3.0 GHz and the missile is illuminated by a nose-on incident vertically polarized plane wave. The overall dimensions of the missile are 2.83m by 0.64m by 0.64m, given an electrical size of 28.3λ by 6.4λ by 6.4λ. The total number of tetrahedron for this model is 3,711,809 using a mesh size of 0.1λ. The simulation uses 64 subdomains and $\xi = 10^{-3}$ as the tolerance

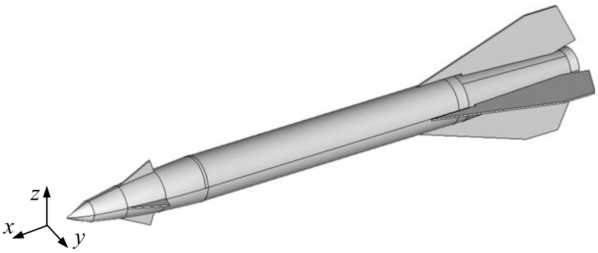
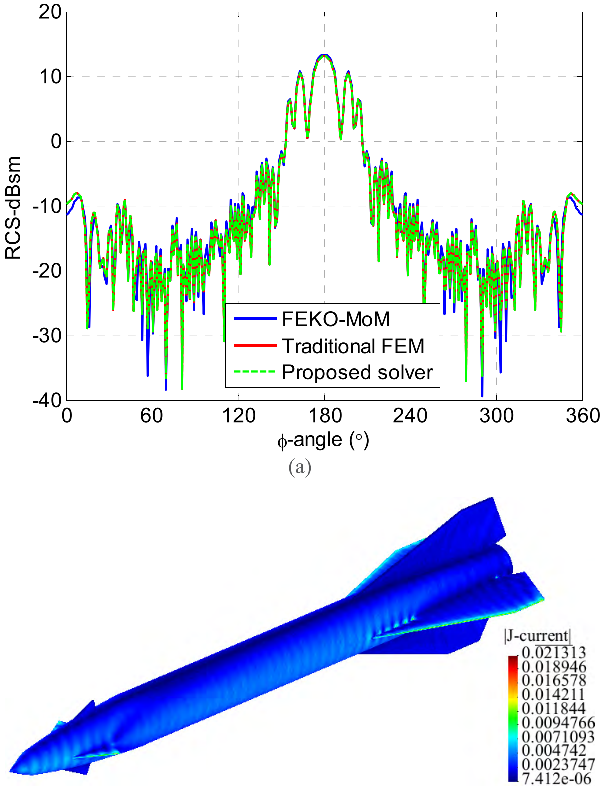


FIGURE 10. Metal missile model.



(a)

(b)

FIGURE 11. Calculation results of metal missile model :(a) Bistatic RCS of xoy plane. (b) Distribution of the electric current (A/m) over the missile surface.

for the iterative GMRES solver. As the previous simulation, the mesh truncation technique employed was the first order ABC.

Figure. 11 (a) shows the comparison of the bistatic RCS result for the elevation cut (XOY plane) given by the traditional FEM version of the solver and the commercial software FEKO. As the previous test, a very good agreement is appreciated. The electric current distribution over the missile is plotted in Fig. 11 (b).

The computational statistics of this test are given in Table 2. The proposed solver only takes 738.6 seconds to complete the simulation using 2 compute nodes (128 CPU cores). However, for the traditional FEM case, although 1280 CPU cores are used, the total simulation time is 2,313.9 seconds, which is about 3 times slower than the proposed methodology.

TABLE 2. Computational statistics of proposed solver and traditional Fem for the modern missile model.

Methods	CPU cores	Unknowns	Iterations with ξ $=10^{-3}$	Peak Memory Usage/GB	Total CPU time/s
Traditional FEM	1280	23,682,848	/	2280.6	2313.9
Proposed	128	25,535,316	163	381.9	738.6
FEKO-MoM	32	133,890	/	268.6	58,134.7
Proposed	32	25,535,316	163	318.2	2125.7

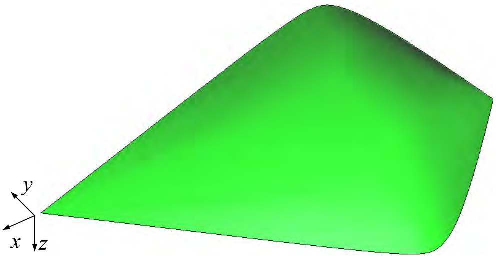


FIGURE 12. The metal low scattering carrier model.

Also, the peak memory usage reaches about 2.3 TB, which is usually difficult to satisfy, even using medium size HPC platforms.

FEKO simulation is performed using its MoM module on a workstation with 80 CPU cores and 1 TB of RAM. The mesh size is set to $0.125\lambda_0$ leading to 89,260 triangles and 133,890 unknowns (note that these unknowns are MoM unknowns). Solving this example using the in-core MoM complex double precision module requires 268.6 GB memory. A total of 32 CPU cores are used, taking 16.1 hours to complete the simulation in the mentioned workstation. For comparison, the simulation of proposed solver is execute using the same computing resources on this workstation. As it happens with HFSS, the proposed solver is also much faster and efficient than FEKO. It is worth noting that we have compared the pure MoM module of FEKO without any acceleration technique such as MLFMA.

3) A LOW SCATTERING OBJECT

This third simulation consists of the analysis of a PEC low-scattering object as the one shown in Fig. 12. Here the truncation of the computational domain is performed using the FE-IIEE method. In this way, we can validate the accuracy and correctness of this technique combined with DDM by comparing the results with those given by an in-house higher order method of moments (HOMoM) [36] and the commercial software FEKO. It is worth noting that the accurate RCS analysis of low-scattering objects is a challenge for FEM truncation techniques due to the low-level values that these objects present. Thus, this example is a good test to show the capabilities of the FE-IIEE algorithm as truncation method.

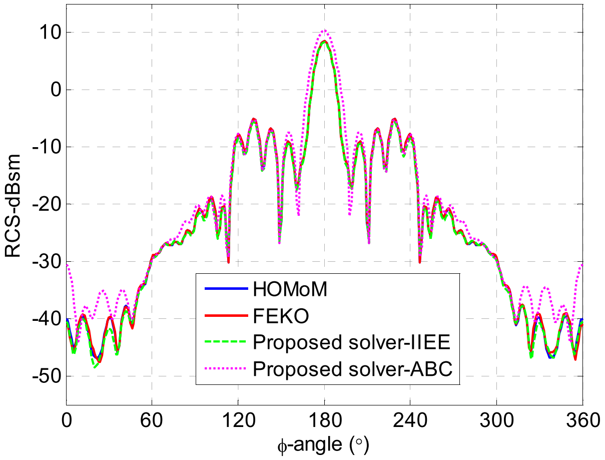


FIGURE 13. Bistatic RCS calculation results for the xoy plane of the low scattering carrier model.

The model is illuminated by a y-axis polarized plane wave propagating along the negative x-axis direction at 3.0 GHz. The electrical size of the model at this frequency is 7.31λ by 5.24λ by 0.77λ. The truncation boundary is set at a distance of 0.25λ from the target. The discretization of the model contains 535,267 tetrahedrons that implies 3,625,506 unknowns. The model is divided into 8 subdomains setting the iterative solver tolerance to $\xi = 10^{-4}$. As commented during the explanation of the FE-IIEE method, the error in the residual of the radiation boundary condition, ψ, is limited by the tolerance of the iterative solver. Thus, in this case, ψ is also set to 10^{-4}. The simulation is performed using 128 CPU cores in total.

Figure 13 shows the bistatic RCS for the XOY plane. It is worth noting that the results given by our FEM solver using the FE-IIEE method as truncation technique, the mentioned HOMoM code and FEKO presents a very good agreement between them. Additionally, the results given by the FEM solver using the first order ABC are also attached in Fig. 13. As it is well-known, these results confirm that the ABC cannot reach accurate values when the truncation boundary is close to the target. However, these results confirm that accuracy of the FE-IIEE method is not related to the location of the external boundary as expected.

Table 3 summarizes the computational statistics of this example when the proposed solver uses the first order ABC and the FE-IIEE method. It can be concluded that, although the FE-IIEE require more computing time than ABC due to its iterative methodology, it provides a robust way to eliminate the mesh truncation error.

B. PARALLEL PERFORMANCE

Once the accuracy validation benchmark is successfully finished, the second benchmark has consisted of the analysis of the parallel performance of the solver. In this case, we have examined the effect of the number of subdomains on the peak memory usage and the parallel scalability in a small number of CPU cores using the previous models. On the other hand, the last test has examined the parallel performance in a large number of CPU cores using an eight times core scale (from 1280 to 10,240).

TABLE 3. Computational statistics of proposed solver using different mesh truncation technology for the low scattering object.

Methods	Scattering field iteration with 10^{-4}	DDM iteration with ξ=10^{-4}	Peak Memory Usage/GB	Total CPU time/s
Proposed-ABC	/	73	39.47	97.3
Proposed-IIEE	9	73/55/36/26/18/ 11/5/2/1	41.78	345.2

TABLE 4. Computing resources and execution time information for the Waveguide Wide-Edge Slot Antenna.

CPU cores	CPU time/s (8domains)	CPU time/s (16domains)	CPU time/s (32domains)	CPU time/s (64domains)
64	1325.5	908.3	539.9	371.3
128	698.0	492.7	286.0	192.0
256	366.3	269.2	155.2	107.0
512	197.9	151.5	89.1	64.8

1) WAVEGUIDE WIDE-EDGE SLOT ANTENNA

This test has consisted of the simulation of the waveguide slot antenna shown in Fig. 8 varying the number of subdomains from 2 to 128. The number of cores employed in the simulations has also been increased from 64 to 512 cores. All other parameters were set as previously.

The total memory usage of all processes when the number of subdomains changes is shown in Fig. 14. All the simulations used 128 cores (2 compute nodes of the supercomputer). A clear behavior is appreciated, that is, when the number of domains increases, the peak memory used by the solver gradually decreases. This behavior provides an advantage when solving larger problems using thousands of CPU cores. However, the memory reduction rate becomes slower when the number of subdomains is too high.

In order to provide a better explanation to this behavior, the matrix forms that are solved when partitioning the original problem into 2 and 4 subdomains are given in Fig.15 respectively. It can be clearly seen that the size of self-domain matrix $\boldsymbol{A}_{ii}^{D(4)}$ when the number of domains is set to 4 is almost half of the size of the matrix $\boldsymbol{A}_{ii}^{D(2)}$ when the number of domains is set to 2. As mentioned in Section II. B, these self-domain matrices are factorized independently using the sparse matrix equation direct solver and then, their factor matrices are kept for subsequent iterative calculations. It is well known that a large amount of fill-in will occur during the sparse matrix decomposition process. In the case shown in Fig. 15, the matrix $\boldsymbol{A}_{11}^{D(4)}$ and $\boldsymbol{A}_{22}^{D(4)}$ can be approximately considered as two sub-matrix blocks of the matrix $\boldsymbol{A}_{11}^{D(2)}$, so their fill-in will be greatly reduced compared to the decomposition of the

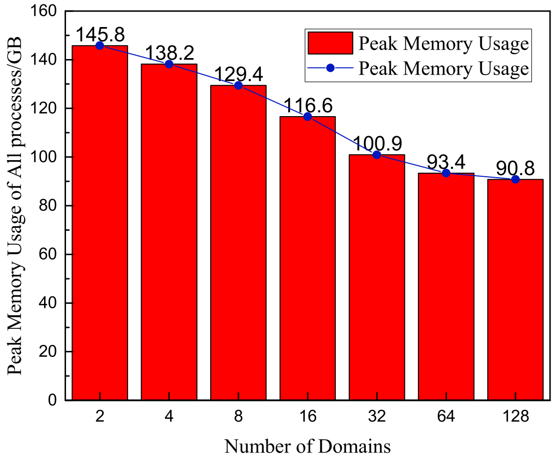


FIGURE 14. **Total peak memory usage of the solver using different number of subdomains for the waveguide slot antenna model.**

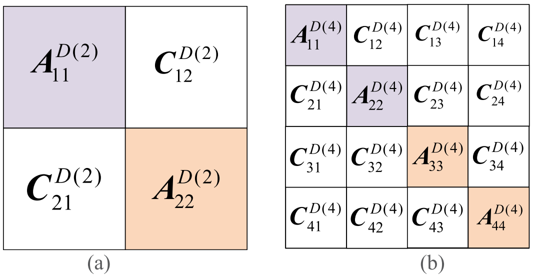


FIGURE 15. **The matrix forms that solved in the proposed solver when partitioning original problem into (a) 2 subdomains (b) 4 subdomains.**

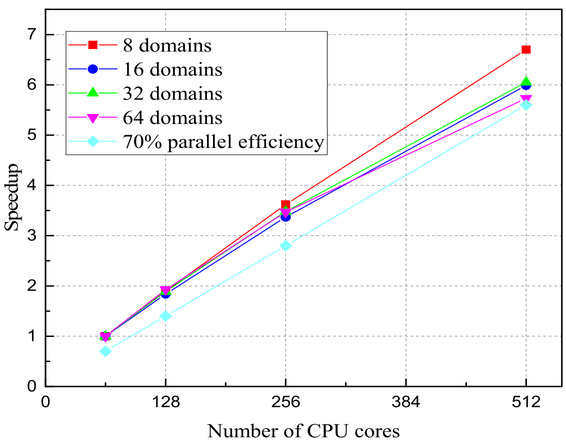


FIGURE 16. **Parallel scalability of the FEM solver when using different number of subdomains for the waveguide slot antenna model.**

large matrix $\boldsymbol{A}_{11}^{D(2)}$ based on the fact that they are completely independently solved. But on the other hand, we need more memory to store the coupling matrix $\boldsymbol{C}_{ij}$ and related basic function information on the interface of adjacent subdomains, when the number of domains is set to 4.

Therefore, when the number of domains reaches a certain level, the size of the subdomain matrix will become too tiny in comparison with the size of the full problem leading to a reduction on the fill-in. Additionally, as the number of domains increases, the storage requirements of the coupling matrix will increase. Thus, the peak memory used by the solver gradually decreases when the number of domains increases. The test result is in line with theoretical expectations. Even when the number of domains continues to increase, it is possible to show an increase of the peak memory consumption, instead of a decrease. Thus, it is important to choose an appropriate number subdomain that ensure high computational efficiency and low storage requirements.

On the other hand, it should be pointed out that the memory reduction is quite limited when the number of domains doubles. This is mainly because the model being solved in here is aperiodic, so each domain's self-domain matrix is completely different and needs to be decomposed. If the model is a periodic structure in which a domain is often a translational or rotational invariance of a building block, it is only necessary to compute correspond matrix for each building block, rather than each domain, thus the memory will drop significantly as presented in most of already published papers [6]–[8].

Another important feature that must to be checked during the parallel performance analysis is the parallel scalability. In this test, the number of domains has been varied from 8 to 64 and the number of CPU cores from 64 to 512 (8 compute nodes of the supercomputer). Table 4 gathers the computing times of each simulation for this test. As shown in Table 4, the CPU time also gradually decreases when the number of domains increases. However, it doesn't mean that the proposed solver will be much more efficient as an element-level domain decomposition method since, as the number of domains increases, the iterative convergence of the DDM will be slower. This behavior has been already presented in most of the published papers [6]–[8]. Figure 16 plots the speedup of each subdomain division, where a parallel efficiency higher than 70% is obtained for all the cases.

2) MODERN MISSILE

The second test has consisted of the simulation of the missile varying the number of subdomains from 4 to 160 and the number of cores from 128 to 1024. In this test, the memory usage and the parallel scalability have also been evaluated.

Figure 17 shows the peak memory behavior of all processes when the missile is divided from 4 to 160 subdomains. All the simulations were performed using 256 cores (4 compute nodes of the supercomputer). In this case, the same memory behavior applies where the peak memory gradually decreases when the number of subdomains increases. However, looking at the memory usage from 128 to 160 subdomains, one may see how the memory reaches the lower value for this simulation (287 GB). As commented previously, the tiny size of each subdomain compared with the size of the full problem makes impossible to keep reducing the memory.

Table 5 gathers the computing time of each simulation for this test where the number of subdomains varies from 16 to 128 and the number of cores varies from 128 to 1024.

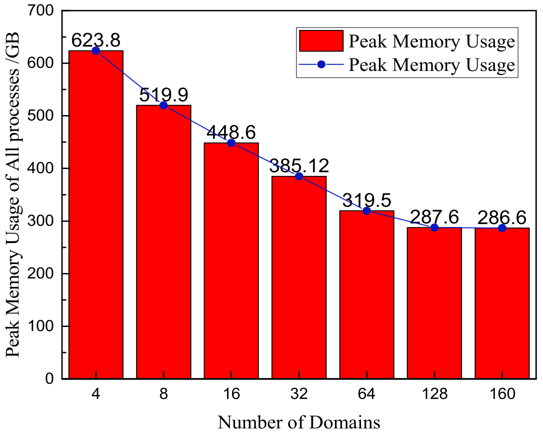


FIGURE 17. Total peak memory usage of the solver using different number of subdomains for the PEC missile model.

TABLE 5. Computing resources and execution time information for the modern missile.

CPU cores	CPU time/s (16domains)	CPU time/s (32domains)	CPU time/s (64domains)	CPU time/s (128domains)
128	3189.4	1729.3	802.8	547.6
256	1616.0	883.2	418.4	284.7
512	822.0	458.9	230.9	151.2
1024	513.1	256.8	150.3	103.3

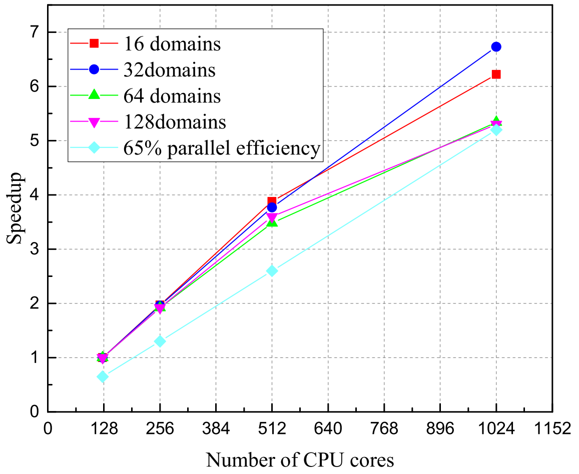


FIGURE 18. Parallel scalability of the FEM solver when using different number of subdomains for the PEC missile model.

Figure 18 plots the speedup of each subdomain division, where a parallel efficiency higher than 65% is obtained. It is important to remark that, in this case, the parallel efficiency when the number of subdomains is greater than 64 experiments a drastically reduction when the number of CPU tends to 1024. This is due to the same reason than the peak memory reaches the limit value when using 160 subdomains: the tiny size of each subdomain in comparison with the size of the full problem. At this point, it is worth to remark that, typically, a numerical code with a parallel efficiency between 55-60 % is considered acceptable. Thus, the proposed solver exceeds this barrier by much demonstrating its massively parallel feature.

TABLE 6. Computing resources and execution time information for the low-scattering object.

CPU cores	Scattering field iteration with 10^{-4}	DDM iteration with $\xi = 10^{-4}$	CPU time/s
72	8	83/63/42/27/17/ 9/3/1	7560.7
144			3924.2
288			2190.4
576			1388.9

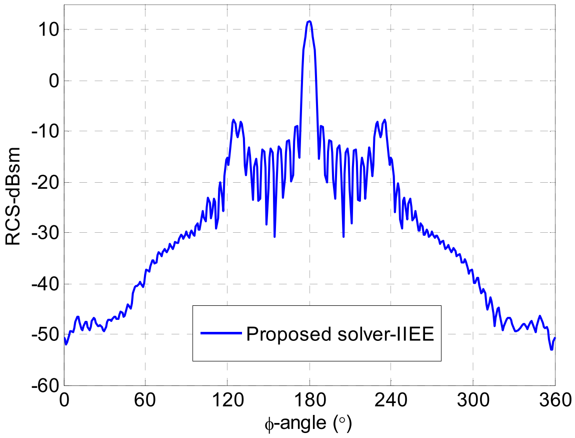


FIGURE 19. Bistatic RCS calculation results for the xoy plane of the low scattering carrier model at 6 GHz.

3) A LOW SCATTERING OBJECT

The third test has consisted of the low-scattering object varying the number of CPU cores from 72 to 576. We perform this test is mainly amid to show the parallel performance of the proposed solver when the FE-IIEE technique is employed. Its simulation was performed at 6 GHz. The discretization of the model contains 4,240,038 tetrahedrons with 29,265,102 unknowns. The model is divided into 64 subdomains setting the iterative solver tolerance to 10^{-4}. In particular, this test is performed in a small-size HPC cluster equipped with 8 compute nodes. Each node has four 18-core Intel(R) Xeon(R) Gold 6140 CPU @ 2.30 GHz and 512 GB RAM.

Figure 19 shows the bistatic RCS for the XOY plane. Table 6 gathers the computing time of each simulation for this test where the number of cores varies from 72 to 576. According to Table 6 the speedup is calculated and plotted in Fig. 20. It can be observed that higher than 65% of the parallel efficiency can also be obtained when the solver using the IIEE method.

4) PARALLEL SCALABILITY OVER 10,000 CPU CORES

Previous tests have demonstrated the parallel capabilities of the solver when using low-medium HPC platforms (up to 1024 CPU cores). However, when solving nowadays

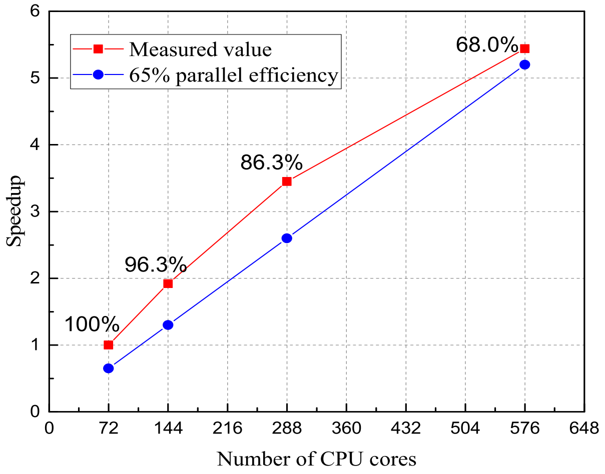


FIGURE 20. Parallel scalability of the FEM solver with activated IIEE for the low scattering object.

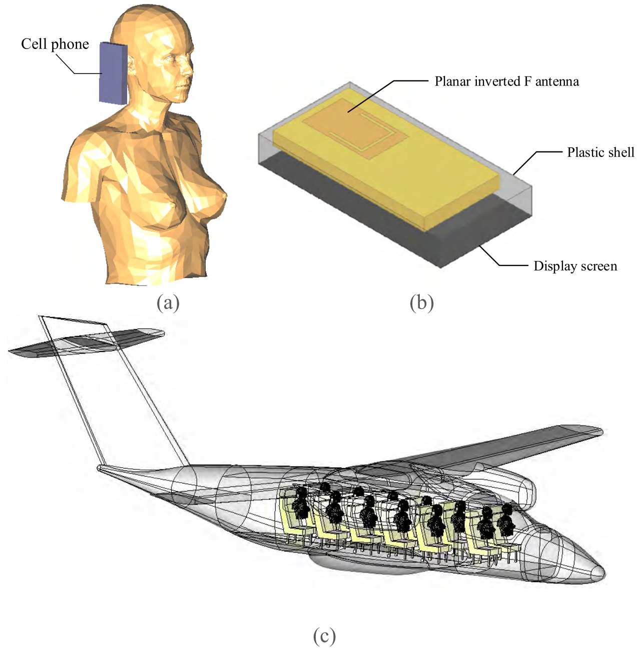


FIGURE 21. Electromagnetic analysis of human body exposed to cell phones in an aircraft :(a) human body model. (b) cell phone model. (c) aircraft with passengers simulation model.

challenging electromagnetic problems, the number of CPU cores employed in the simulation, typically, exceeds 10,000 cores. In this situation, load balancing and communication among processes are very difficult to control and a very stable and very efficient code is required to extract all the computational power of the supercomputer.

The calculation scenario for this test has consisted of the electromagnetic analysis of the human body exposed to cell phones in an aircraft. The female body model is shown in Fig. 21 (a). The cell phone displayed in Fig. 21 (b) is placed about 2 cm away from right ear of the female model. The aircraft model is filled with 12 female bodies, each one with the corresponding mobile phone except the pilots. The cell phone antenna operates at 0.9 GHz and has an input power of 0.25 W. The three-dimensional dimensions of the aircraft are 14.0 m × 16.1 m × 4.2 m, and the corresponding electrical dimensions are 42.0 $\lambda \times$ 48.3 $\lambda \times$ 12.6 λ. Relative permittivity of the female body is 45.5 and the conductivity is 0.7. Relative permittivity of the seat inside the aircraft is 1.1 and the dielectric loss tangent is 0.0075. The number of tetrahedron generated for this model is 32,624,124, and the corresponding number of unknowns is 222,191,512. The model is divided into 320 subdomains.

TABLE 7. Computing resources and execution time information for the aircraft model.

CPU cores	Unknowns	Number of Domains	Iterations with $\xi=5\times10^{-3}$	CPU time/ Excitation/(s)
1280	222,191,512	320	263	6662.1
2560				3490.2
5120				2273.8
10,240				1347.4

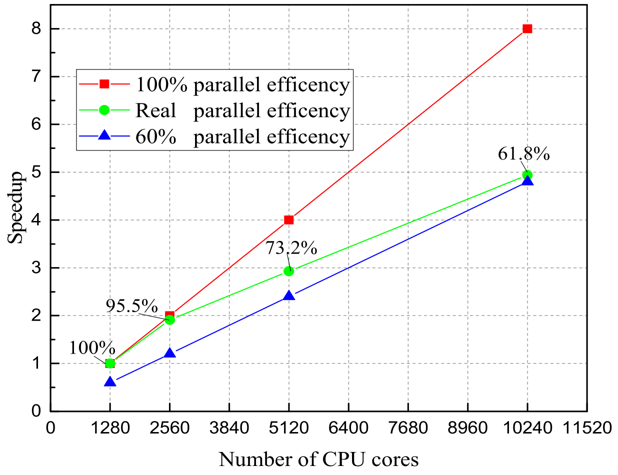


FIGURE 22. Parallel scalability of the FEM solver when using 10,240 CPU cores for the aircraft with passengers model.

Table 7 gathers the statistics and computing time of each simulation for this test. According to this data, the corresponding parallel speedup curve and the parallel efficiency values are plotted in Fig. 22. A parallel efficiency higher than 60% is obtained using up to 10,240 CPU cores (160 compute nodes of the supercomputer). This 60% efficiency is an excellent value for such a large computing scale, demonstrating the good scalability and implementation of the solver. Figure 23 show the near field distribution in some places of the plane.

C. CHALLENGING ELECTROMAGNETIC PROBLEM

Finally, the simulation of a challenging electromagnetic problem is carried out. This simulation has consisted of the analysis of a waveguide wide-edge slot antenna composed by 64 elements and 4096 slots (Fig. 24). The antenna element

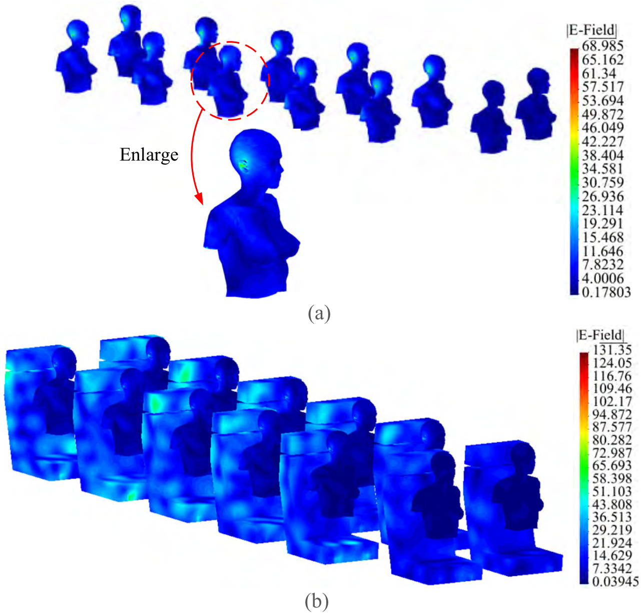


FIGURE 23. The electric filed distribution at :(a) passengers' bodies (b) passengers' bodies and seats.

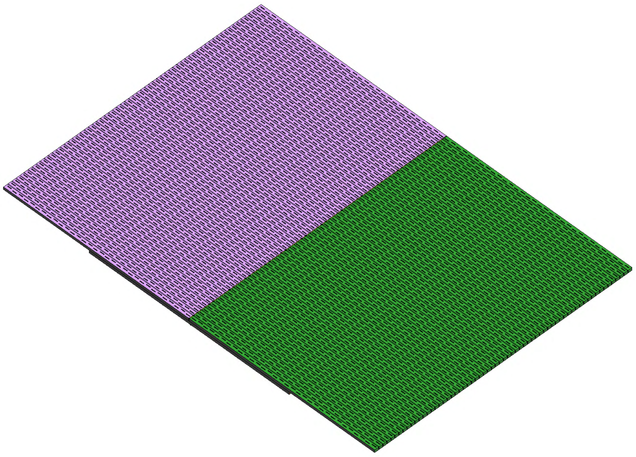


FIGURE 24. Large waveguide wide-edge slot antenna array model.

is the waveguide previously analyzed (Fig. 8). The operating frequency is 18.7 GHz. The electrical size of the antenna at this frequency is 32.2λ by 47.6λ. The total number of tetrahedron for this antenna is 45,378,548, and the number of unknowns is 307,519,378. The antenna array is divided into 400 subdomains. The iterative tolerance is set to $\xi = 10^{-3}$.

A total of 12,800 CPU cores (200 compute nodes of the supercomputer) were used to perform this simulation. The simulation took 1070 minutes and 9.39 TB of memory using 64 excitations. The number of iterations of the GMRES solver for each excitation was approximately 170 steps. The radiation pattern for two elevation cuts is shown in Fig. 25. It confirms that the proposed method can efficiently perform full-wave simulation of challenging electromagnetic problems.

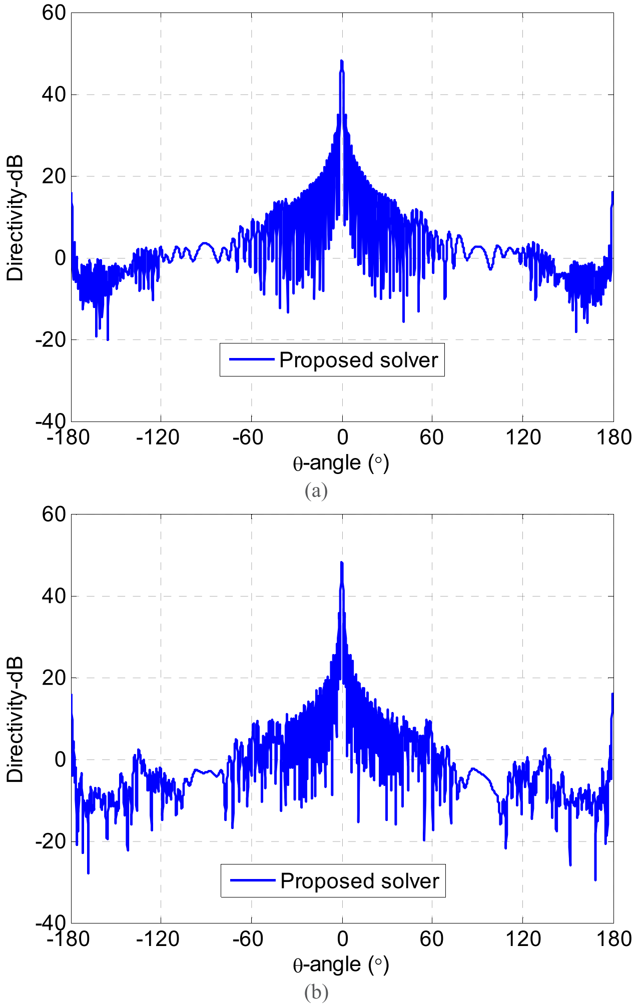


FIGURE 25. The radiation pattern of large waveguide wide-edge slot antenna array at :(a) xoz plane. (b) yoz plane.

V. CONCLUSION

An efficient massively parallel finite element method solver for the solution of complex and electrically large electromagnetic problems with arbitrary structures is presented. The solver makes use of a domain decomposition algorithm to decomposes the original problem into several non-overlapping subdomains and a numerical exact mesh truncation algorithm called finite element-iterative integral equation evaluation (FE-IIEE). Thanks to these techniques the solver is able to provide very accurate solutions to challenging scattering and antenna problems. Compared to previously published massively parallel FEM codes, the proposed method comprehensively considers the calculation accuracy and computational efficiency, and can reduce RAM memory peaks and accelerate the speedups of computation more effectively while making all relevant calculation precisions controllable. Different tests have demonstrated the parallel capabilities of the solver when using low-medium HPC platforms (up to 1024 CPU cores) and large HPC platforms (up to 10,240 CPU cores) obtaining always parallel efficiencies above 60 % and parallel speedup above 4.8 times (the parallel scale is expanded by 8 times). Finally, a 64 elements large waveguide wide-edge slot antenna array model discretized into 45,378,548 tetrahedrons is analyzed efficiently using 12,800 CPU cores demonstrating the power of the presented solver.

ACKNOWLEDGMENT

Authors would like to thank Sugon company for providing access to the computing resources employed in this paper. We wish to thank the support staff for their hard work.

REFERENCES

[1] O. G. Ernst and M. J. Gander, "Why it is difficult to solve Helmholtz problems with classical iterative methods," in *Numerical Analysis of Multiscale Problems*. Berlin, Germany: Springer, 2012, pp. 325–363.

[2] *MUMPS Solver*. Accessed: 2018. [Online]. Available: http://mumps.enseeiht.fr/index.php?page=home

[3] *MKL PARDISO Solver*. Accessed: 2018. [Online]. Available: https://software.intel.com/en-us/articles/intel-mkl-pardiso

[4] B. Despres, P. Joly, and J. E. Roberts, "A domain decomposition method for the harmonic Maxwell equations," in *Iterative Methods Linear Algebra*. Amsterdam, The Netherland: North Holland, 1992, pp. 475–484.

[5] M. N. Vouvakis, Z. Cendes, and J.-F. Lee, "A FEM domain decomposition method for photonic and electromagnetic band gap structures," *IEEE Trans. Antennas Propag.*, vol. 54, no. 2, pp. 721–733, Feb. 2006.

[6] K. Zhao, V. Rawat, S.-C. Lee, and J.-F. Lee, "A domain decomposition method with nonconformal meshes for finite periodic and semi-periodic structures," *IEEE Trans. Antennas Propag.*, vol. 55, no. 9, pp. 2559–2570, Sep. 2007.

[7] M. Vavoukis and J. F. Lee, "Modeling large finite antenna arrays with non-conformal, non-overlapping finite-element domain-decomposition methods," *URSI Radio Sci. Bull.*, vol. 84, no. 1, pp. 35–58, 2011.

[8] Z. Peng and J.-F. Lee, "Non-conformal domain decomposition method with mixed true second order transmission condition for solving large finite antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 59, no. 5, pp. 1638–1651, May 2011.

[9] M.-F. Xue and J.-M. Jin, "Nonconformal FETI-DP methods for large-scale electromagnetic simulation," *IEEE Trans. Antennas Propag.*, vol. 60, no. 9, pp. 4291–4305, Sep. 2012.

[10] Z. Peng, K.-H. Lim, and J.-F. Lee, "Nonconformal domain decomposition methods for solving large multiscale electromagnetic scattering problems," *Proc. IEEE*, vol. 101, no. 2, pp. 298–319, Feb. 2013.

[11] J. Ma, J.-M. Jin, and Z. Nie, "A nonconformal FEM-DDM with tree–cotree splitting and improved transmission condition for modeling subsurface detection problems," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 355–364, Jan. 2014.

[12] V. Dolean, M. J. Gander, and S. Lanteri, "Effective transmission conditions for domain decomposition methods applied to the time-harmonic curl–curl Maxwell's equations," *J. Comput. Phys.*, vol. 280, pp. 232–247, Jan. 2015.

[13] W.-J. Wang *et al.*, "Massively parallel simulation of large-scale electromagnetic problems using one high-performance computing scheme and domain decomposition method," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 5, pp. 1523–1531, Oct. 2017.

[14] P.-H. Jia, J. Hu, Y. Chen, R. Zhang, L. Lei, and Z. Nie, "H-matrices compressed multiplicative Schwarz preconditioner for nonconformal FEM-BEM-DDM," *IEEE Trans. Antennas Propag.*, vol. 66, no. 5, pp. 2691–2696, May 2018

[15] B. Michiels, J. Fostier, I. Bogaert, and D. D. Zutter, "Full-wave simulations of electromagnetic scattering problems with billions of unknowns," *IEEE Trans. Antennas Propag.*, vol. 63, no. 2, pp. 796–799, Feb. 2015.

[16] X. M. Pan, W.-C. Pi, M.-L. Yang, Z. Peng, and X.-Q. Sheng, "Solving problems with over one billion unknowns by the MLFMA," *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2571–2574, May 2012.

[17] W. Gropp, T. Hoefler, and R. Thakur, *Using Advanced MPI: Modern Features of the Message-Passing Interface*. Cambridge, MA, USA: MIT Press, 2014.

[18] R. Chandra, L. Dagum, and D. Kohr, *Parallel Programming in OpenMP*. San Mateo, CA, USA: Morgan Kaufmann, 2001.

[19] B. Engquist and A. Majda, "Absorbing boundary conditions for numerical simulation of waves," *Math. Comput.*, vol. 31, no. 5, pp. 421–435, 1977.

[20] J.-Y. Wu, D. M. Kingsland, J.-F. Lee, and R. Lee, "A comparison of anisotropic PML to Berenger's PML and its application to the finite-element method for EM scattering," *IEEE Trans. Antennas Propag.*, vol. 45, no. 1, pp. 40–50, Jan. 1997.

[21] X. Q. Sheng and Z. Peng, "Analysis of scattering by large objects with off-diagonally anisotropic material using finite element-boundary integral-multilevel fast multipole algorithm," *IET Microw., Antennas Propag.*, vol. 4, no. 4, pp. 492–500, Apr. 2010.

[22] R. Fernandez-Recio, L. E. Garcia-Castillo, I. Gomez-Revuelto, and M. Salazar-Palma, "Convergence study of a non-standard Schwarz domain decomposition method for finite element mesh truncation in electromagnetics," *Prog. Electromagn. Res.*, vol. 120, no. 1, pp. 439–457, 2011.

[23] I. Gomez-Revuelto, L. E. Garcia-Castillo, and L. F. Demkowicz, "A comparison between PML, infinite elements and an iterative BEM as mesh truncation methods for HP self-adaptive procedures in electromagnetics," *Prog. Electromagn. Res.*, vol. 126, no. 5, pp. 499–519, 2012.

[24] D. Garcia-Donoro, L. E. Garcia-Castillo, T. K. Sarkar, and Z. Yu, "A non-standard Schwarz domain decomposition method for finite element mesh truncation of infinite arrays," *IEEE Trans. Antennas Propag.*, to be published. doi: 10.1109/TAP.2018.2866532.

[25] A. Tzoulis and T. F. Eibert, "Efficient electromagnetic near-field computation by the multilevel fast multipole method employing mixed near-field/far-field translations," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, pp. 449–452, 2005.

[26] J. C. Nédélec, "Mixed finite elements in $\mathbb{R}^3$," *Numer. Math.*, vol. 35, no. 3, pp. 315–341, 1980.

[27] L. E. García-Castillo and M. Salazar-Palma, "Second-order Nédélec tetrahedral element for computational electromagnetics," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 13, nos. 2–3, pp. 261–287, Mar./Jun. 2000.

[28] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, 1986.

[29] *SuperLU Solver*. Accessed: 2018. [Online]. Available: http://crd-legacy.lbl.gov/~xiaoye/SuperLU/

[30] S. Alfonzetti, G. Borzi, and N. Salerno, "Iteratively-improved Robin boundary conditions for the finite element solution of scattering problems in unbounded domains," *Int. J. Numer. Methods Eng.*, vol. 42, no. 4, pp. 601–629, 1998.

[31] M. Vouvakis, K. Zhao, S.-M. Seo, and J.-F. Lee, "A domain decomposition approach for non-conformal couplings between finite and boundary elements for unbounded electromagnetic problems in R^3," *J. Comput. Phys.*, vol. 225, no. 1, pp. 975–994, 2007. doi: 10.1016/j.jcp.2007.01.014.

[32] X. Zhao, S.-W. Ting, and Y. Zhang, "Parallelization of half-space MLFMA using adaptive direction partitioning strategy," *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 1203–1206, 2014.

[33] *ParMetis*. Accessed: 2018. [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview

[34] *HFSS*. Accessed: 2018. [Online]. Available: https://www.ansys.com/Products/Electronics/ANSYS-HFSS

[35] *FEKO*. Accessed: 2018. [Online]. Available: https://altairhyperworks.com/product/FEKO

[36] T. K. Sarkar, *Higher Order Basis Based Integral Equation Solver (HOBBIES)*. Hoboken, NJ, USA: Wiley, 2012.

SHENG ZUO was born in Hunan, China, in 1992. He received the B.S. degree in electronic information engineering and the M.S. degree in electronics and communications engineering from Xidian University, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include the large-scale parallel finite element computation in complex electromagnetic environment, domain decomposition method, and fast multipole method.

DANIEL GARCÍA DOÑORO (M'14) was born in Madrid, Spain, in 1983. He received the Degree of Telecommunication Engineering and the Ph.D. degree in multimedia and communications from the University Carlos III of Madrid, in 2008 and 2014, respectively. From 2008 to 2009, he was a Visiting Research Scholar with Syracuse University, where he was involved in 3D modeling applied to electromagnetic problems. Since 2016, he has been an Associate Professor with Xidian University, Xi'an, China. He has authored or co-authored two book, 15 articles in international journals, and 44 articles in international conferences, symposiums, and workshops. His research interests include the numerical methods to high-performance computational electromagnetics, including finite elements, domain decomposition methods, and hp-adaptivity.

YU ZHANG received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 1999, 2002, and 2004, respectively. In 2004, he joined Xidian University as a Faculty Member. He was a Visiting Scholar and an Adjunct Professor with Syracuse University, from 2006 to 2009. As a Principal Investigator, he works on projects, including the Project of NSFC. He has authored four books: *Parallel Computation in Electromagnetics* (Xidian Univ. Press, 2006), *Parallel Solution of Integral Equation-Based EM Problems in the Frequency Domain* (Wiley–IEEE, 2009), *Time and Frequency Domain Solutions of EM Problems Using Integral Equations and a Hybrid Methodology* (Wiley, 2010), and *Higher Order Basis Based Integral Equation Solver* (Wiley, 2012), as well as more than 100 journal articles and 40 conference papers.

YANG BAI was born in Shaanxi, China, in 1992. She received the B.S. degree in electronic information engineering from Sichuan University, in 2014. She is currently pursuing the M.S. degree with Xidian University. Her research interests include the finite element electromagnetic modeling and finite element thermal simulation theory.

XUNWANG ZHAO received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2004 and 2008, respectively. In 2008, he joined Xidian University as a Faculty Member. As a Principal Investigator, he is doing or has completed some projects, including the Project of NSFC.