Journal of Computational Physics 228 (2009) 3255–3267

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Parallel implementation of the FETI-DPEM algorithm for general 3D EM simulations

Yu-Jia Li, Jian-Ming Jin *

Center for Computational Electromagnetics, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801-2991, United States

ARTICLE INFO

Article history:
Received 22 July 2008
Received in revised form 4 December 2008
Accepted 13 January 2009
Available online 6 February 2009

Keywords:
Domain decomposition method (DDM)
Dual-primal method
Finite element tearing and interconnecting (FETI)
Parallel computation

ABSTRACT

A parallel implementation of the electromagnetic dual-primal finite element tearing and interconnecting algorithm (FETI-DPEM) is designed for general three-dimensional (3D) electromagnetic large-scale simulations. As a domain decomposition implementation of the finite element method, the FETI-DPEM algorithm provides fully decoupled subdomain problems and an excellent numerical scalability, and thus is well suited for parallel computation. The parallel implementation of the FETI-DPEM algorithm on a distributed-memory system using the message passing interface (MPI) is discussed in detail along with a few practical guidelines obtained from numerical experiments. Numerical examples are provided to demonstrate the efficiency of the parallel implementation.

1. Introduction

Engineering simulations of large and complex structures require the development of fast algorithms which can fully utilize parallel computing resources. Domain decomposition-based methods, which can provide a balanced data distribution across processors, become the best choice for such simulations. Among a variety of domain decomposition methods (DDMs), the dual-primal finite element tearing and interconnecting (FETI-DP) method exhibits an excellent numerical scalability and has become one of the most scalable parallel solvers in computational mechanics and acoustics [1–5].

As the edge-element implementation of the FETI-DP method for electromagnetic analysis, the FETI-DPEM method has been proposed recently for the simulation of three-dimensional (3D) open-region electromagnetic problems [6–9]. The FETI-DPEM method is shown to be numerically scalable [8] by combining the dual-primal (DP) idea with two Lagrange multipliers and implementing a Robin-type transmission condition at the subdomain interfaces. Over the past few years, the serial implementation of the FETI-DPEM method has been successfully applied to finite array problems by fully exploiting their geometrical repetitions. The first parallel implementation of the FETI method for electromagnetic analysis was explored in [10] without any coarse grid correction. The convergence rate of the FETI-H method [11] is recently studied in [12] for solving electromagnetic scattering problems, where "H" stands for the Helmholtz equation. A relatively simple OpenMP implementation was carried out for a specialized FETI-DPEM algorithm and used for simulating finite array problems [13]. For other DDMs applied in electromagnetics, an accurate solution and a fast convergence cannot be obtained simultaneously due to the use of auxiliary variables defined at the edges shared by faces with discontinuous normal directions [14]. An acceptable convergence rate is achieved only with a compromised accuracy and by using the Gauss-Seidel preconditioner,

* Corresponding author. Tel.: +1 217 244 0756; fax: +1 217 333 5962.
E-mail address: j-jin1@uiuc.edu (J.-M. Jin).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.01.029

which has been found inefficient for parallel computation. In contrast, the numerical scalability is achieved in the FETI-DPEM without using any preconditioner.

The general principle of the FETI-DPEM method is first to divide the entire computational domain into nonoverlapping subdomains, where an incomplete solution of the field in the subdomain is constructed using a direct solver. Next, tangential field continuities are enforced through the Robin-type transmission condition at the subdomain interfaces by using two Lagrange multipliers. This yields an equivalent reduced-order interface problem, which can be solved using an iterative algorithm. This iterative solution using the Krylov subspace method is then accelerated by propagating the residual error to the whole computational domain at each iteration through the construction of a global coarse problem related to the degrees of freedom (DOFs) at the subdomain corner edges (edges shared by more than two subdomains interior to the computational domain). The solution to the interface problem can then serve as the boundary condition for individual subdomain problems to evaluate the field inside the subdomains. The aforementioned process is highly parallelizable and can be implemented for the simulations of large-scale electromagnetic problems using massively parallel systems.

In this paper, the FETI-DPEM algorithm developed in [8] is outlined first. It is followed by the detailed discussion of its parallel implementation on a distributed-memory system using the message passing interface (MPI). Finally, numerical examples are then provided to demonstrate the efficiency of the implementation.

2. The FETI-DPEM formulation

The FETI-DPEM method, as the edge-element implementation of the FETI-DP method [1–5], has been developed for electromagnetic analysis in [6], in which its potential capability for parallel computing is theoretically explored. To make the FETI-DPEM method scalable on a massively parallel computing system, a numerically scalable algorithm was developed in [8] by constructing a coarse grid correction at subdomain corner edges and utilizing the Robin-type transmission condition at the subdomain interfaces to replace the original Dirichlet transmission condition used in [6]. In this section, the FETI-DPEM method introduced in [8] is outlined to provide necessary information for subsequent discussions.

In the FETI-DPEM method, a computational domain is first divided into nonoverlapping subdomains. In the ith subdomain, the electric field at the subdomain interfaces and the subdomain-level corner related system can be obtained after eliminating the subdomain interior volumetric unknowns as

$$\mathbf{B}_r^i E_r^i = \mathbf{B}_r^i \mathbf{K}_{rr}^{i^{-1}} (f_r^i - \lambda_r^i - \mathbf{K}_{rc}^i E_c^i) \tag{1}$$

and

$$\left(\mathbf{K}_{cc}^i - \mathbf{K}_{rc}^{i^T} \mathbf{K}_{rr}^{i^{-1}} \mathbf{K}_{rc}^i\right) \mathbf{B}_c^i E_c = f_c^i - \lambda_c^i - \mathbf{K}_{rc}^{i^T} \mathbf{K}_{rr}^{i^{-1}} f_r^i + \mathbf{K}_{rc}^{i^T} \mathbf{K}_{rr}^{i^{-1}} \lambda_r^i \tag{2}$$

where the subscripts c and r represent the subdomain corner DOFs and the remaining DOFs, respectively. Matrices $\mathbf{K}_{rr}^i$, $\mathbf{K}_{rc}^i$, and $\mathbf{K}_{cc}^i$ and excitation vectors f_r^i and f_c^i are the subdomain finite element system matrices and excitation vectors, respectively. The dual unknown λ represents the contribution from the Robin-type transmission condition at the subdomain interfaces. The Boolean matrix $\mathbf{B}_r^i$ extracts the interface DOFs of the ith subdomain and $\mathbf{B}_c^i$ extracts the local corner DOFs from the global corner DOFs. It is noted that at this stage the subdomain interior fields become fully decoupled and the associated volumetric unknowns are eliminated independently with the introduction of the dual unknown λ at the subdomain interfaces.

To couple the fields over all subdomains, a matrix equation representing the interface continuity condition is then derived by enforcing the electric and magnetic tangential field continuities across the subdomain interfaces through the Robin-type transmission condition. This yields the matrix equation

$$\tilde{\mathbf{K}}_{rr} \lambda + \tilde{\mathbf{K}}_{rc} E_c = \tilde{f}_r \tag{3}$$

where

$$\begin{aligned}
\tilde{\mathbf{K}}_{rr} &= \mathbf{I} + \sum_{i=1}^{N_s} \mathbf{Q}^{i^T} \sum_{j \in \text{neighbor}(i)} \mathbf{T}_j^{i^T} \left(\mathbf{T}_i^j - 2\mathbf{M}_{ij} \mathbf{T}_i^j \mathbf{B}_r^j \mathbf{K}_{rr}^{j^{-1}} \mathbf{B}_r^{i^T}\right) \mathbf{Q}^j \\
\tilde{\mathbf{K}}_{rc} &= -2 \sum_{i=1}^{N_s} \mathbf{Q}^{i^T} \sum_{j \in \text{neighbor}(i)} \mathbf{T}_j^{i^T} \mathbf{M}_{ij} \mathbf{T}_i^j \mathbf{B}_r^j \mathbf{K}_{rr}^{j^{-1}} \mathbf{K}_{rc}^j \mathbf{B}_c^j \\
\tilde{f}_r &= -2 \sum_{i=1}^{N_s} \mathbf{Q}^{i^T} \sum_{j \in \text{neighbor}(i)} \mathbf{T}_j^{i^T} \mathbf{M}_{ij} \mathbf{T}_i^j \mathbf{B}_r^j \mathbf{K}_{rr}^{j^{-1}} f_r^j
\end{aligned} \tag{4}$$

where the Boolean matrix $\mathbf{Q}^i$ extracts from the dual unknowns λ the component in the ith subdomain and $\mathbf{T}_j^i$ extracts the interface DOFs associated with Γ^{ij} in the ith subdomain, where Γ^{ij} denotes the interface shared by adjacent subdomains i and j. The mass-like matrix $\mathbf{M}_{ij}$ is a primal-to-dual projection matrix between the ith and jth subdomains. The reader is referred to [8] for their detailed expressions.

The global corner DOFs related system equation is obtained by assembling the subdomain contributions from (2), and can be written as

$$\tilde{\mathbf{K}}_{cc} E_c = \tilde{f}_c + \tilde{\mathbf{K}}_{cr} \lambda \tag{5}$$

where

$$\begin{aligned}
\tilde{\mathbf{K}}_{cc} &= \sum_{i=1}^{N_s} \tilde{\mathbf{K}}_{cc}^i = \sum_{i=1}^{N_s} \left[\mathbf{B}_c^{iT} \mathbf{K}_{cc}^i \mathbf{B}_c^i - \left(\mathbf{K}_{rc}^i \mathbf{B}_c^i \right)^T \mathbf{K}_{rr}^{i^{-1}} \left(\mathbf{K}_{rc}^i \mathbf{B}_c^i \right) \right] \\
\tilde{\mathbf{K}}_{cr} &= \sum_{i=1}^{N_s} \left(\mathbf{B}_r^i \mathbf{K}_{rr}^{i^{-1}} \mathbf{K}_{rc}^i \mathbf{B}_c^i \right)^T \mathbf{Q}^i \\
\tilde{f}_c &= \sum_{i=1}^{N_s} \tilde{f}_c^i = \sum_{i=1}^{N_s} \left(\mathbf{B}_c^{iT} f_c^i - \mathbf{B}_c^{iT} \mathbf{K}_{rc}^{iT} \mathbf{K}_{rr}^{i^{-1}} f_r^i \right).
\end{aligned} \tag{6}$$

The matrix $\tilde{\mathbf{K}}_{cc}$ is a highly sparse symmetric matrix representing the global corner DOFs related system.
By combining (3) and (5) and eliminating E_c, we obtain the FETI-DPEM interface equation for the dual unknowns λ

$$\left[\tilde{\mathbf{K}}_{rr} + \tilde{\mathbf{K}}_{rc} \tilde{\mathbf{K}}_{cc}^{-1} \tilde{\mathbf{K}}_{cr} \right] \lambda = \tilde{f}_r - \tilde{\mathbf{K}}_{rc} \tilde{\mathbf{K}}_{cc}^{-1} \tilde{f}_c. \tag{7}$$

The interface Eq. (7) is generally indefinite, and can be solved using a Krylov subspace method. After the interface problem is solved, the electric field inside each subdomain can be evaluated independently by using the calculated boundary condition λ at the subdomain interfaces.

3. Parallel implementation of the FETI-DPEM algorithm

The scalability of the FETI and FETI-DP algorithms for various engineering applications using massively parallel computation has been explored and demonstrated in [1–5,15,16]. In this paper, we focus on the parallel implementation of the FETI-DPEM algorithm on a distributed-memory system using the MPI for 3D electromagnetic analysis.

3.1. Parallel implementation framework

Before applying the FETI-DPEM method, we first decompose the computational domain into N_s subdomains automatically using a greedy-like algorithm and a deterministic optimization scheme provided by an automatic meshing partition software DOMDEC [17]. In general, the subdomain interfaces resulting from such a decomposition typically have irregular shapes, multi-scale mesh sizes, and material inhomogeneities. The basic requirements for an effective decomposition are such that it provides a small number of nodes on subdomain interfaces and a balanced load for parallel computation. A balanced load

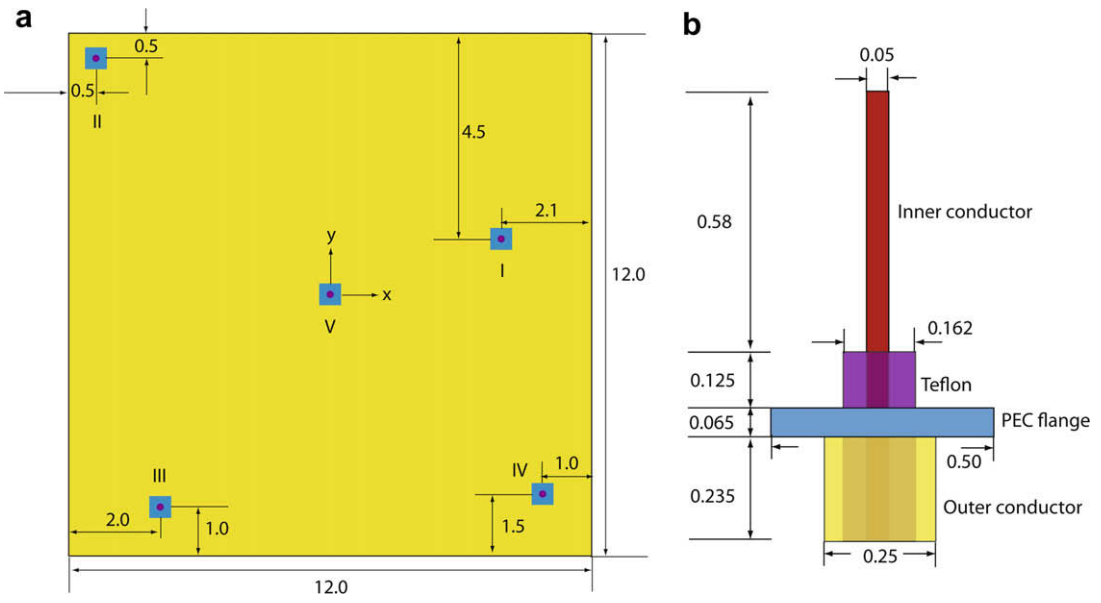

Fig. 1. Geometry of five monopoles on a finite ground plane. All dimensions are in inches. (a) Top view. (b) Side view of a monopole with SMA connector.

here refers to a similar number of unknowns for each subdomain, which yields a similar amount of work for each processor. The decomposed subdomain data are distributed across processors, which can then be processed independently and concurrently in parallel.

The implementation of the FETI-DPEM algorithm consists of three major steps: the preprocessing step, the solution of the interface Eq. (7), and the recovery of subdomain solutions. In the preprocessing step, each subdomain problem is processed, where the subdomain finite element system matrices $\mathbf{K}_{rr}^i, \mathbf{K}_{rc}^i, \mathbf{K}_{cc}^i$ and $\mathbf{M}_{ij}$ and the excitation vectors f_r^i and f_c^i are assembled. In addition, the sparse matrix $\mathbf{K}_{rr}^i$ is factorized, whose factorization is stored and will be used repeatedly throughout the simulation. As mentioned in Section 2, such a preprocessing procedure only involves subdomain operations; thus, it can be carried out completely in parallel. The preprocessing step is followed by the iterative solution of the interface Eq. (7), where the information exchanges among processors occur in the matrix–vector multiplication process at each iteration. During the matrix–vector multiplication process, the calculation of the subdomain contributions can be performed completely in parallel by using the information obtained in the preprocessing step, whose details are provided in Section 3.2. The global corner DOFs related system $\tilde{\mathbf{K}}_{cc}$ is fully assembled over all subdomains in parallel. Such a corner system is relatively small due to the small number of corner DOFs. Hence, a local copy of its factorization is kept by all processors to reduce the parallel overhead and implementation complexity. Detailed discussions about the effective ways to implement the parallel solution of the coarse problem are provided in Section 3.3. After solving the interface problem, the fields in the subdomains are then recovered independently and concurrently in parallel by using the calculated boundary condition at the subdomain interfaces.

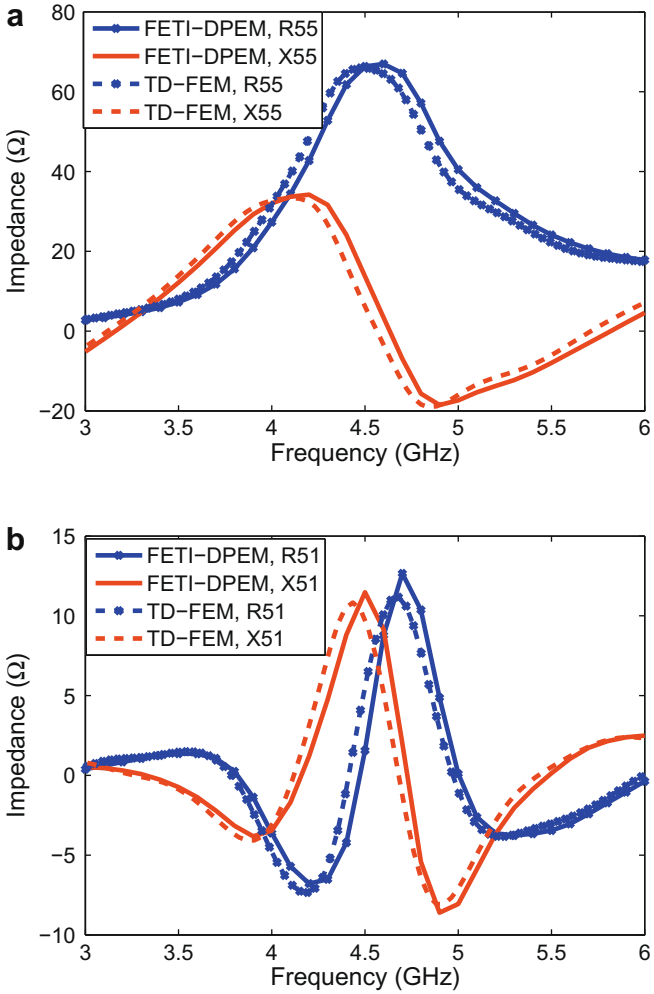

Fig. 2. Input and mutual impedances of monopole V, where R and X are the resistance and reactance, respectively. (a) Input impedance $Z_{55} = R_{55} + jX_{55}$. (b) Mutual impedance $Z_{51} = R_{51} + jX_{51}$.

3.2. Parallel matrix–vector multiplication

When using an iterative method to solve (7), we need to evaluate the matrix–vector multiplication $\delta^k = \left[\tilde{\mathbf{K}}_{rr} + \tilde{\mathbf{K}}_{rc}\tilde{\mathbf{K}}_{cc}^{-1}\tilde{\mathbf{K}}_{cr}\right]\lambda^k$ at the kth iteration. To start with, we store λ^k and the local matrix-vector product δ^k in a distributed fashion to avoid the parallel bottleneck when λ^k becomes prohibitively large. More specifically, each processor holds a segment of λ^k as input and also outputs its own share of δ^k at the end of the matrix–vector multiplication process. This calculation can be accomplished in two steps:

Step 1: Calculate $\delta^k = \tilde{\mathbf{K}}_{rr}\lambda^k = \lambda^k + \sum_{i=1}^{N_s}\mathbf{Q}^{i^T}\sum_{j\in\text{neighbor}(i)}\mathbf{T}_j^{i^T}\left(\mathbf{T}_i^j - 2\mathbf{M}_{ij}\mathbf{T}_i^j\mathbf{B}_r^j\mathbf{K}_{rr}^{j^{-1}}\mathbf{B}_r^{j^T}\right)\mathbf{Q}^j\lambda^k$.
Step 2: Calculate $\delta^k = \delta^k + \tilde{\mathbf{K}}_{rc}\tilde{\mathbf{K}}_{cc}^{-1}\tilde{\mathbf{K}}_{cr}\lambda^k$.

Using the distributed λ^k, all the required computations in the jth subdomain including subdomain matrix–vector multiplications and the solution of the subdomain local problems in Step 1 can be performed subdomain-by-subdomain, and thus can be easily parallelized. The communications between the neighboring subdomains are required when implementing the $\sum_{j\in\text{neighbor}(i)}\mathbf{T}_j^{i^T}$ operator, which requires a sequence of data exchanges between two processors representing two adjacent subdomains. These data exchanges are realized by non-blocking communications *isend* and *irecv*, which also allow the overlap of communication and computation. In order to carry out Step 2, we split it into three substeps:

Step 2-1: Calculate $y^k = \tilde{\mathbf{K}}_{cr}\lambda^k = \sum_{i=1}^{N_s}\left(\mathbf{B}_r^i\mathbf{K}_{rr}^{i^{-1}}\mathbf{K}_{rc}^i\mathbf{B}_c^i\right)^T\mathbf{Q}^i\lambda^k$.
Step 2-2: Solve $\tilde{\mathbf{K}}_{cc}x^k = y^k$ for x^k.
Step 2-3: Calculate $\delta^k = \delta^k + \tilde{\mathbf{K}}_{rc}x^k = \delta^k - 2\sum_{i=1}^{N_s}\mathbf{Q}^{i^T}\sum_{j\in\text{neighbor}(i)}\mathbf{T}_j^{i^T}\mathbf{M}_{ij}\mathbf{T}_i^j\mathbf{B}_r^j\mathbf{K}_{rr}^{j^{-1}}\mathbf{K}_{rc}^j\mathbf{B}_c^jx^k$.

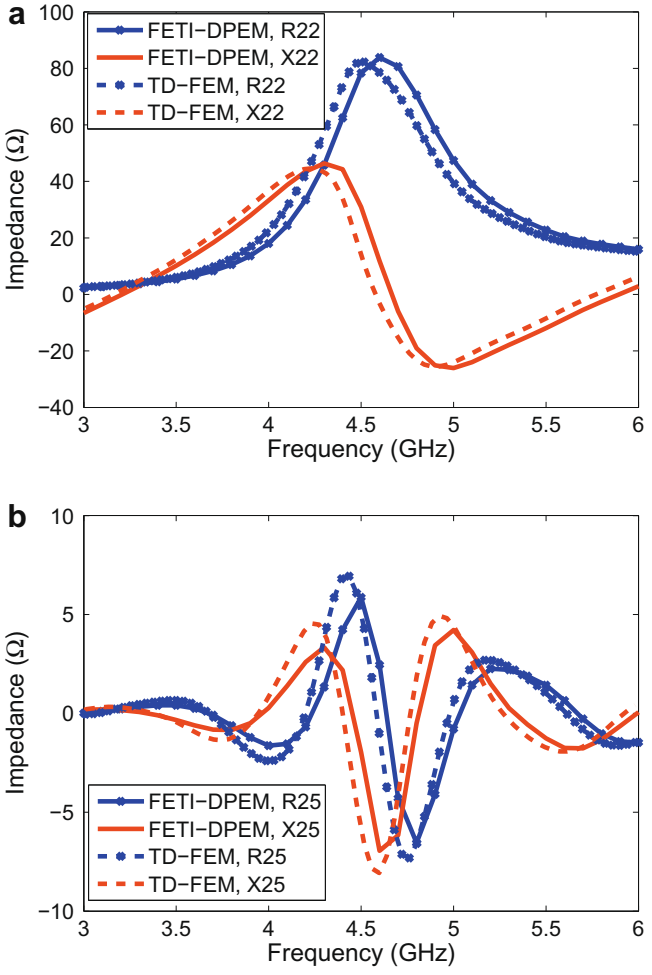

Fig. 3. Input and mutual impedances of monopole *II*, where *R* and *X* are the resistance and reactance, respectively. (a) Input impedance $Z_{22} = R_{22} + jX_{22}$. (b) Mutual impedance $Z_{25} = R_{25} + jX_{25}$.

Using the distributed λ^k, the calculations in Step 2-1 only involve subdomain-level operations and can be fully parallelized, whose local sums are *reduced* to obtain y^k. The solution in Step 2-2 is treated as a coarse grid correction, which couples all the subdomain computations and increases the convergence rate by propagating the error globally at each iteration. More implementation details about Step 2-2 are provided in Section 3.3. The calculated x^k is *scattered* to all processors in Step 2-3, where the subdomain-level calculations can be carried out in parallel and the information exchanges between the neighboring subdomains are accomplished in the same non-blocking communications performed in Step 1.

When using the parallel conjugate gradient (CG) based iterative solver, two extra synchronizations are required to implement the matrix–vector multiplication process due to the existence of the global coarse problem. It will be shown in Section 3.3 that the number of synchronizations can be reduced by carefully manipulating the solution to the coarse problem. It is worthwhile mentioning that $\mathbf{B}_r, \mathbf{B}_c, \mathbf{Q}$ and $\mathbf{T}$ are Boolean matrices, whose matrix–vector multiplications can be performed without any floating-point operation.

3.3. Implementation of the coarse problem

Step 2-2 in Section 3.2 is referred to as the coarse grid correction, which increases the convergence by damping the low-frequency component error in the solution similar to the one in the multigrid method. As mentioned in Section 3.2, such a coarse problem is global, which unfortunately increases the complexity in the parallel implementation. Three effective ways to implement the solution of the coarse problem are described in [1–5] [15] and [16], which can be summarized as follows.

(1) In the first implementation, a local copy of $\tilde{\mathbf{K}}_{cc}$'s factorization is kept by all the processors. Such a factorization is used repeatedly in the forward and backward substitutions in the solution of the coarse problem in Step 2-2, which reduces the parallel overhead by decreasing the number of required synchronizations. Instead of introducing two extra synchronizations in Step 2, only one *allreduce* process is required to obtain x^k from all processors and then redistribute it. However, it is worthy mentioning that the factorization of $\tilde{\mathbf{K}}_{cc}$ is accomplished completely in serial, which decreases the parallel speedup as the number of processors increases according to Amdahl's law.
(2) The second implementation partitions the full matrix $\tilde{\mathbf{K}}_{cc}^{-1}$ such that each processor keeps the related submatrix of $\tilde{\mathbf{K}}_{cc}^{-1}$. Therefore, a matrix–vector multiplication is carried out instead of performing forward and backward substitutions in Step 2-2 for x^k. The benefit of this implementation is that only one synchronization *allreduce* is required in Step 2. However, such an implementation shares the same disadvantage as the first implementation and it is typically less efficient when the number of iterations for solving the interface problem is small.
(3) The third implementation applies a parallel sparse direct solver to the coarse problem. In this case, the $\tilde{\mathbf{K}}_{cc}$ matrix is first factorized in parallel and parallel forward and backward substitutions are carried out at each iteration. Such an implementation minimizes the CPU idle time but it increases the parallel overhead and the number of synchronizations at each iteration and also requires a highly effective parallel sparse solver even for very small problems.

As a summary, the optimal implementation of the coarse problem requires the trade-off between the distribution of floating-point operations and its related parallel overhead; thus, it is typically machine and problem dependent. In this work, all the implementations have been tested. Due to the small coarse problem sizes for all the testing cases, it has been found that the first implementation outperforms in all numerical experiments; therefore, only the results using the first implementation are provided.

3.4. Iterative solver for the interface problem

In the previous implementation of FETI-DPEM method, the restarted generalized minimum residual (GMRES) method is applied to solve the indefinite and unsymmetric interface Eq. (7), which provides a fast and monotonically decreasing con-

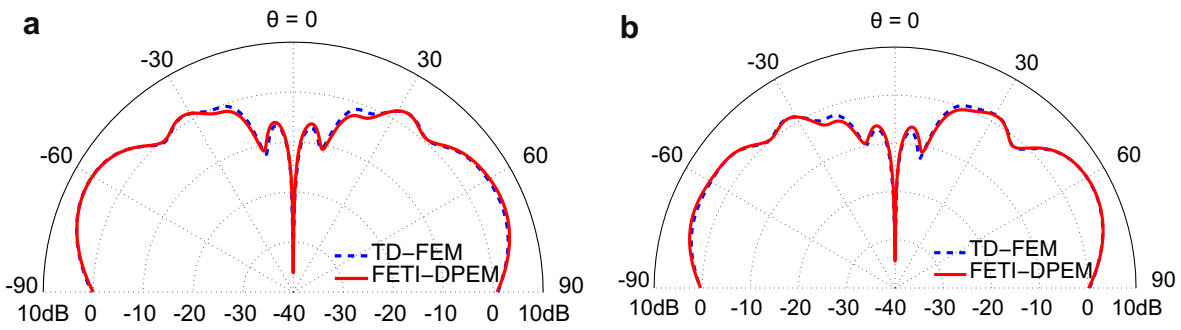

Fig. 4. Gain patterns in two principal planes at 4.6 GHz when only monopole *V* is excited. (a) $\phi = 0$ plane. (b) $\phi = \pi/2$ plane.

vergence [6–8]. However, the restarted GMRES suffers from the poor scalability in parallelization [18] and hence, is replaced by a CG based method such as the stabilized biconjugate gradient (BiCGSTAB) method in this work for a cheaper memory usage, a comparable convergence, and a better parallel scalability. When applying the BiCGSTAB method to the iterative solution of the interface Eq. (7), two matrix–vector products are calculated in parallel at each iteration. Other vector computations such as the calculations of the residue and search direction vectors can also be performed on a subdomain-by-subdomain basis; thus, they are also parallelized. More specifically, the computation of vector inner products is performed by each processor and the local partial sums are accumulated and redistributed by the global communication *allreduce*.

3.5. Using higher-order hierarchical basis functions

Higher-order hierarchical basis functions [19] are employed in the FETI-DPEM for its easy p-adaption and a better condition number in the resultant finite element matrix. These basis functions are formed by adding new functions to the lower-order basis functions and also provide separate representation of the gradient and rotational parts of the vector field. In the FETI-DPEM method, the enforcement of the field continuity is through the equalization of DOF pairs with the aid of the Lagrange multipliers. When using the lowest-order hierarchical basis function, such DOF pairs are associated with the finite element edges at the subdomain interfaces, which can be easily identified using the global coordinates associated with edge vertices. However, when higher-order hierarchical basis functions are used, the face basis functions are introduced and multiple DOF pairs are associated with each edge, which makes the identification of DOF pairs extremely difficult. It is noticed

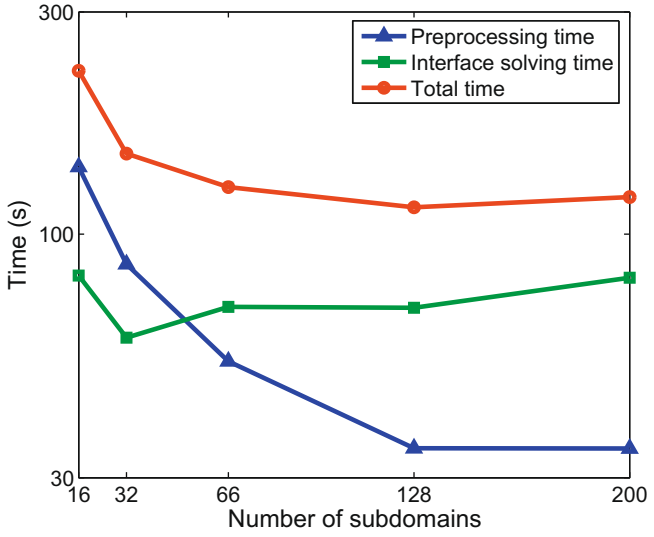

Fig. 5. Wall clock time spent on the individual parts of the FETI-DPEM solution for different numbers of subdomains using 16 processors.

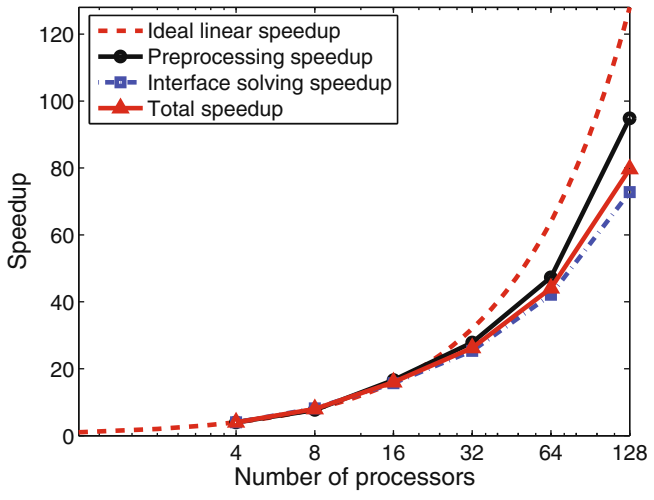

Fig. 6. Parallel speedup versus the number of processors with $N_s = 128$.

that if a face (triangle) located at the subdomain interfaces has the same local node numbering from different subdomains, the DOF pairs can be identified easily using the global coordinate information associated with the vertices of each face. Such a regulation is accomplished by reordering the local nodes in each element based on their global coordinates. As a by-product, it also eliminates the regulation process to align the directions of the basis functions associated with the subdomain interfaces.

4. Numerical examples

In this section, we validate the parallel implementation of the FETI-DPEM algorithm and demonstrate its capability by simulating general 3D open-region electromagnetic problems. All the computations were performed on an Apple Xserve sys-

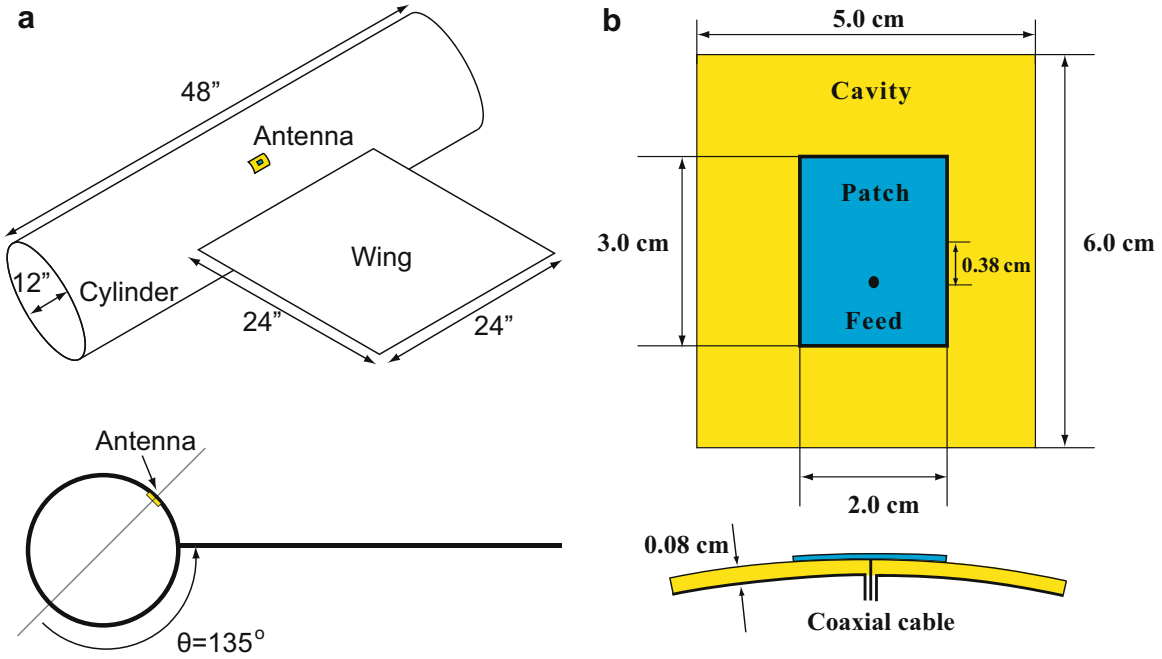

Fig. 7. A patch antenna radiating on a platform. (a) Geometry of the platform. (b) Geometry of the patch antenna.

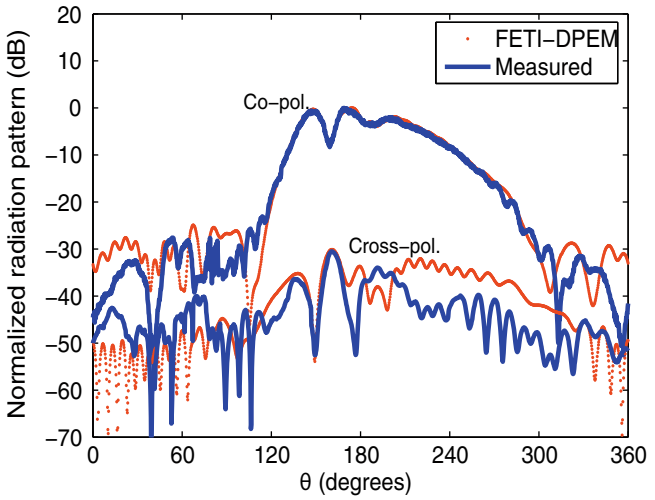

Fig. 8. *H*-plane radiation patterns at 3.3 GHz for a patch antenna on a cylinder with a wing.

tem (cluster distributed-memory architecture), where each node has two 2 GHz G5 processors and 4 GB of RAM. For each simulation, the BiCGSTAB method with a convergence criterion of $\varepsilon < 1 \times 10^{-3}$ is used for solving the interface equation.

The first example is taken from the Electromagnetic Code Consortium (EMCC) antenna benchmarks [20], in which the radiation of five identical monopoles with SMA connectors placed on a finite ground plane is analyzed using the FETI-DPEM method. The detailed configuration of the monopole array and the geometry of a monopole with the SMA connector are shown in Fig. 1, where the teflon in the coaxial cable has a dielectric constant of 2 and a loss tangent of 0.0002. In the calculation, the monopoles are excited one at a time, while others are terminated with a matched load. The self impedances Z_{55} and Z_{22} for monopoles *V* and *II*, calculated at the input port of the SMA connectors, are shown in Figs. 2 and 3 together with the calculated mutual impedances Z_{51} and Z_{25}. Good agreement has been observed over a wide frequency band with the results obtained using the time-domain finite element method (TD-FEM) [21]. The $\hat{\theta}$-polarized gain patterns when monopole *V* is excited at 4.6 GHz are shown in Fig. 4 for the two principal planes: $\phi = 0$ and $\phi = \pi/2$, and compared with the TD-FEM results.

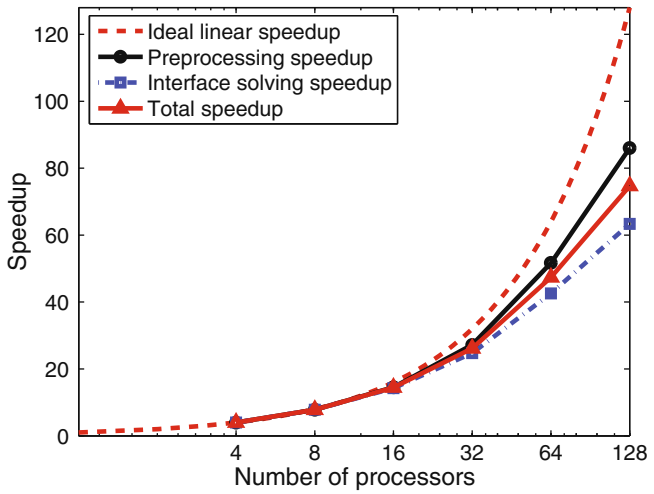

Fig. 9. Parallel speedup versus the number of processors with $N_s = 128$.

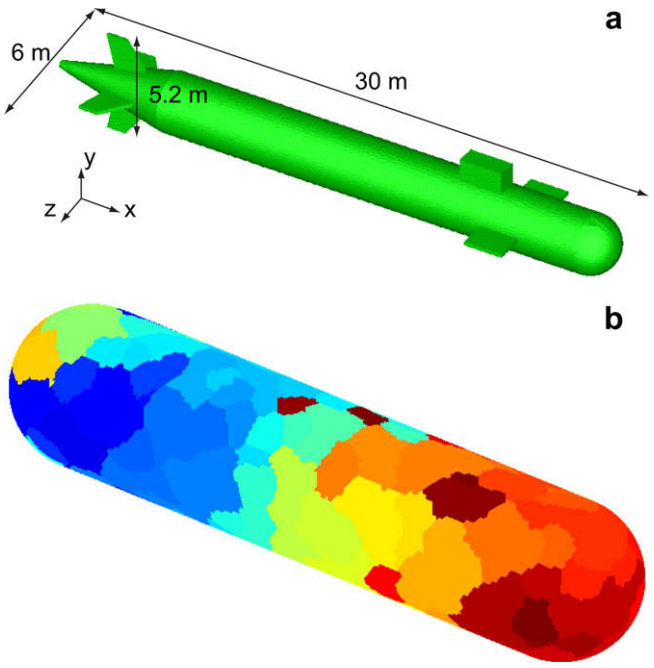

Fig. 10. Simplified submarine structure simulated. (a) Geometry of a free-standing submarine. (b) Illustration of the computational domain decomposed into 514 subdomains.

In the DDM process, it is obvious that, as the number of subdomain increases, the time associated with the subdomain analysis in the preprocessing step deceases monotonically due to the reduced size of the subdomain problems, whereas the time to solve the interface equation first decreases due to smaller subdomain problems and then increases due to an increasing number of interface and corner unknowns. Therefore, in this simulation, we decompose the computational domain into 16, 32, 66, 128 and 200 subdomains to find an optimal decomposition. The wall clock time for calculating the solution at a single frequency using 16 processors is reported in Fig. 5, where the optimal number of subdomains for this problem is found to be $N_s = 128$. In the simulation, a discretization size of $\lambda/h = 10$ and second-order hierarchical basis functions are used, where h is the mesh size. When using this optimal number of subdomains, the total number of FEM unknowns, the size of the interface equation, and the size of the coarse problem are $1.4 \times 10^6, 1.9 \times 10^5$ and 3.7×10^3, respectively. This corresponds to approximately 11,000 unknowns for each subdomain. It can be seen in Fig. 5 that the total computational time is not very sensitive to the number of subdomains around the optimal number. A general rule to determine the number of subdomains is to divide the total number of FEM unknowns by a number between $7,000 \sim 25,000$, which can be used to determine the optimal number of subdomains for all other problems with no need for any further numerical experiments. Averaged over the whole frequency band, 66 iterations are required to solve the interface equation, and for the reference purpose a total solution time of 1.9 min for each frequency sample is required when 16 processors are used.

As the parallel performance is concerned, the parallel speedup is defined with respect to the wall clock time using four processors as

$$Speedup = \frac{4 \times T_4}{T_{N_p}} \tag{8}$$

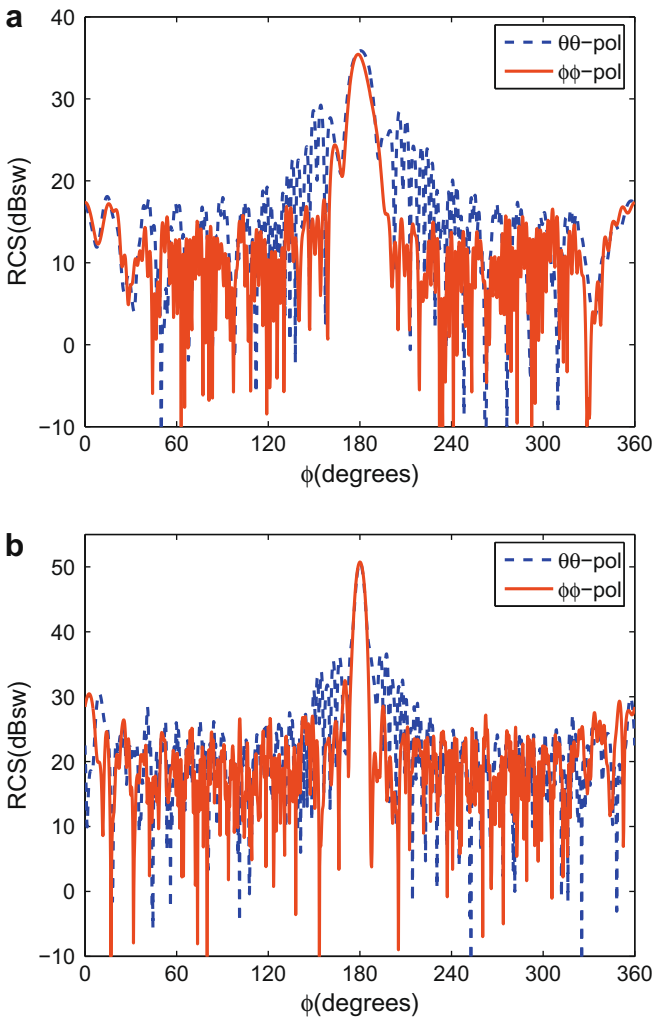

Fig. 11. Bi-static RCS for the submarine in the $x - y$ plane. (a) 300 MHz. (b) 600 MHz.

where T_4 and T_{N_p} denote the wall clock time using four and N_p processors, respectively. The total parallel speedup using up to $N_p = 128$ processors is shown in Fig. 6 together with the parallel speedups for the preprocessing step and the interface solution step, when the number of subdomains is set to $N_s = 128$. It can be observed that an excellent total parallel speedup is achieved when $N_p \leqslant 32$, and a reasonable performance is obtained when N_p is further increased.

In the next example, a conformal cavity-backed patch antenna radiating on a platform composed of a conducting cylinder and a wing is simulated. As shown in Fig. 7, the patch antenna with its longer edge aligned with the cylinder's axis is fed with a coaxial line and housed in a dielectric filled rectangular cavity with $\epsilon_r = 2.17$. The coax feed with filling material of $\epsilon_r = 2.17$ has an inner radius of $r_{in} = 0.6$ mm and an outer radius of $r_{out} = 2.05$ mm. The normalized radiation pattern of the patch antenna in the H-plane is calculated at its resonant frequency of 3.3 GHz and compared in Fig. 8 with measurement data [22,23]. The numerical result agrees well with the measurement for both co- and cross-polarizations. In the simulation, a discretization size of $\lambda/h = 6$ and third-order hierarchical basis functions are used and the optimal decomposition is determined as $N_s = 128$ using the rule stated earlier, which corresponds to approximately 23,000 unknowns for each subdomain. The total number of FEM unknowns, the size of the interface equation, and the size of the coarse problem are 2.9×10^6, 3.8×10^5 and 5.7×10^3, respectively. The parallel speedup versus the number of processors defined in (8) is shown in Fig. 9, where for the reference purpose a total solution time of 4.3 min is required when 16 processors are used.

The two examples above were chosen because there are independent solutions available to validate the solutions of the proposed method. With the current parallel implementation described in this paper, the FETI-DPEM algorithm can deal with much larger general electromagnetic problems than could be handled by its serial version. It is observed in the numerical results that as N_p increases, the parallel efficiency decreases using our current implementation. Such a performance can

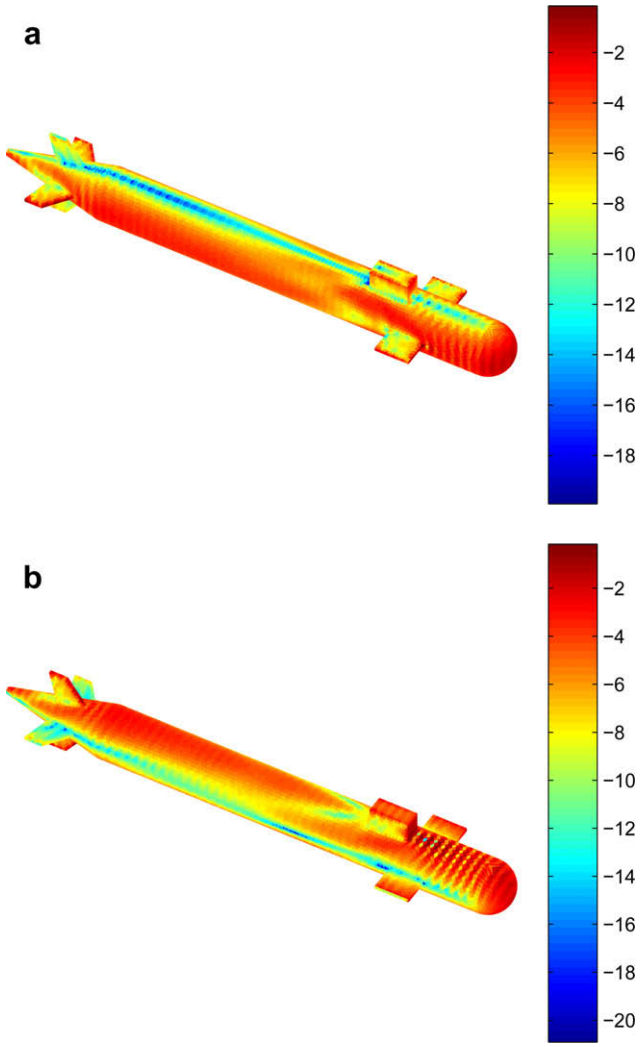

Fig. 12. Electric current induced on the surface of the submarine by a plane wave incident from the $+\hat{x}$ direction. (a) Current distribution from a $\hat{\theta}$-polarized plane wave. (b) Current distribution from a $\hat{\phi}$-polarized plane wave.

Table 1
Number of iterations required for solving the interface equation at different operating frequencies.

Frequency (MHz)	Number of iterations	Frequency (MHz)	Number of iterations
50	74	100	60
150	57	200	50
250	49	300	61
350	48	400	49
450	52	500	50
550	48	600	49
650	49	700	51
750	52	800	53
850	62	900	59
950	68	1000	71

be explained in the following two aspects. First of all, as mentioned in Section 3.3, the global coarse problem is implemented in serial, which includes the sparse matrix factorization and the forward and backward substitutions. Although its associated time is small, it limits the maximum achievable speedup according to Amdahl's law. Unfortunately, this limitation cannot be alleviated easily by using a parallel sparse direct solver due to the sequential nature of the forward and backward substitutions. An alternative solution to this problem is discussed in [3] by assigning the coarse problem to only a small number of processors. The performance for such an implementation is problem dependent and beyond the scope of this paper. Second, for general applications, a perfect load balance for parallel computation is almost impossible to obtain. In this paper, both examples analyzed contain complicated structures such as fine details and thin dielectric substrates, which impose additional difficulty for distributing the work load to each processor. Moreover, it is worthy mentioning that the port boundary condition [24] applied on the coaxial ports makes the mesh decomposition even more tricky, which requires mesh decomposition does not cut through port surfaces. As we can see in the preprocessing speedup curve in Figs. 6 and 9, the load imbalance becomes more and more obvious as N_p increases, which also indicates the upper limits that can be achieved in the parallel computation.

In the last example, the parallelized FETI-DPEM algorithm is applied to the simulation of plane wave scattering by a free-standing submarine-like metallic structure, whose geometry is illustrated in Fig. 10(a). The largest dimension in the $x-$, $y-$ and z-directions is 30 m, 5.2 m and 6 m, respectively. A truncation surface with the first-order absorbing boundary condition is placed 3 m away from the submarine surface. The truncation surface has the shape of a circular cylinder having a radius of 4.5 m and length of 28 m, with both ends capped by a hemispherical surface, yielding a total length of 37 m. For the simulation, a discretization size (the average linear dimension of tetrahedral elements) of $h = 0.2$ m and third-order hierarchical basis functions are used, which yields a total of 1.33×10^7 unknowns. The computational domain is automatically partitioned into 514 subdomains, and the resultant size of the interface equation and the coarse problem is 9.4×10^5 and 2.5×10^4, respectively. The decomposed computational domain is illustrated in Fig. 10(b). The whole structure is excited by a plane wave incident from the $+\hat{x}$ direction. Fig. 11 shows the bi-static radar cross-sections (RCS) in the $x - y$ plane at 300 and 600 MHz for the $\hat{\theta}$ and $\hat{\phi}$ polarizations, respectively. In addition, the induced electric current on the submarine surface at 300 MHz is displayed in Fig. 12. In total, 61 and 49 iterations are required to solve the interface equation at 300 and 600 MHz, respectively, and a total solution time of 7.9 and 7.1 min are required using 64 processors. Finally, a numerical scalability test of the FETI-DPEM algorithm is carried out on this scattering problem to further examine the performance as the operating frequency increases. In the numerical experiment, we vary the operating frequency of the incident field, while fixing the total problem size by fixing the discretization size h, the subdomain size H, and the number of subdomains N_s. In this case, the first version of the FETI-DPEM method is demonstrated to be unscalable [6], whereas the second version provides a numerically scalable result, as observed in [8] for other test problems. Table 1 shows the iteration numbers used to solve the interface equation at different operating frequencies. In the frequency band considered, the mesh density changes gradually from $\lambda/h = 30$ to $\lambda/h = 1.5$. In this case, the FETI-DPEM algorithm shows again the numerical scalability with respect to the working frequency, as observed in [8].

5. Conclusions

This paper presented the parallel implementation of the FETI-DPEM algorithm for general 3D open-region electromagnetic problems on a distributed-memory system using MPI. In such an implementation, the subdomain data are distributed across processors, where the subdomain finite element system matrices are assembled and factorized completely in parallel. Subdomain information is then exchanged among processors in the parallel matrix–vector multiplication process for the iterative solution of the interface equation using the BiCGSTAB method. Finally, fields in the subdomains are recovered in parallel using the calculated boundary condition at the subdomain interfaces. Implementation details were described and practical guidelines were given together with numerical examples showing the performance of the parallel implementation. A relatively good parallel efficiency has been achieved using such an implementation on a cluster system with an automatic decomposition of the computational domain into hundreds of subdomains.

Acknowledgments

This work was partly supported by a grant from the Air Force Office of Scientific Research via the MURI Program under Contract Number FA9550-04-1-0326 and by a grant from the High-Performance Modernization Program of the Department of Defense.

References

[1] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, D. Rixen, FETI-DP: a dual-primal unified FETI method – Part I: A faster alternative to the two-level FETI method, Int. J. Numer. Methods Eng. 50 (2001) 1523–1544.
[2] M. Lesoinne, K. Pierson, FETI-DP: an efficient, scalable and unified dual-primal FETI method, in: Proceedings of the 12th International Conference on Domain Decomposition Methods, 1999, pp. 421–428.
[3] K. Pierson, G. Reese, P. Raghaven, Experiences with FETI-DP in a production level finite element application, in: Proceedings of the 14th International Conference on Domain Decomposition Methods, 2002, pp. 233–240.
[4] C. Farhat, J. Li, P. Avery, A FETI-DP method for the parallel iterative solution of indefinite and complex-valued solid and shell vibration problems, Int. J. Numer. Methods Eng. 63 (2004) 398–427.
[5] C. Farhat, P. Avery, R. Tezaur, J. Li, FETI-DPH: a dual-primal domain decomposition method for acoustic scattering, J. Comput. Acoust. 13 (2005) 499–524.
[6] Y.J. Li, J.M. Jin, A vector dual-primal finite element tearing and interconnecting method for solving 3D large-scale electromagnetic problems, IEEE Trans. Antennas Propagat. 54 (2006) 3000–3009.
[7] Y.J. Li, J.M. Jin, Fast full-wave analysis of large-scale three-dimensional photonic crystal devices, J. Opt. Soc. Am. B 24 (2007) 2406–2415.
[8] Y.J. Li, J.M. Jin, A new dual-primal domain decomposition approach for finite element simulation 3D large-scale electromagnetic problems, IEEE Trans. Antennas Propagat. 55 (2007) 2803–2810.
[9] Y.J. Li, J.M. Jin, Implementation of the 2nd-order ABC with auxiliary variables in the FETI-DPEM method for 3D EM problems, IEEE Trans. Antennas Propagat. 56 (2008) 2765–2769.
[10] C.T. Wolfe, U. Navsariwala, S.D. Gedney, A parallel finite element tearing and interconnecting algorithm for solution of the vector wave equation with PML absorbing medium, IEEE Trans. Antennas Propagat. 48 (2000) 278–284.
[11] C. Farhat, A. Macedo, M. Lesoinne, A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems, Numer. Math. 85 (2000) 283–308.
[12] C.T. Wolfe, S.D. Gedney, Preconditioning the FETI method for accelerating the solution of large EM scattering problems, IEEE Antennas Wireless Propagat. Lett. 6 (2007) 175–178.
[13] Y.J. Li, J.M. Jin, Simulation of photonic crystal nanocavity using the FETI-DPEM method, Microwave Opt. Tech. Lett. 50 (2008) 2083–2086.
[14] V. Rawat, J.-F. Lee, Treatment of cement variables in the domain decomposition method for Maxwells equations, IEEE APS Int. Symp. Dig. (2007) 5937–5940.
[15] M. Lesoinne, K. Pierson, An efficient FETI implementation on distributed shared memory machines with independent numbers of subdomains and processors, Contemp. Math. 218 (1998) 318–324.
[16] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, D. Rixen, Application of the FETI method to ASCI problems – scalability results on 1000 processors and discussion of highly heterogeneous problems, Int. J. Numer. Methods Eng. 47 (2000) 513–535.
[17] C. Farhat, S. Lanteri, H.D. Simon, TOP/DOMDEC – a software tool for mesh partitioning and parallel processing, Comput. Syst. Eng. 6 (1995) 13–26.
[18] R.D. da Cunha, T.R. Hopkins, A parallel implementation of the restarted GMRES iterative method for nonsymmetric systems of linear equations, Adv. Comput. Math. 2 (1994) 261–277.
[19] J.P. Webb, Hierarchical vector basis functions of arbitrary order for triangular and tetrahedral finite elements, IEEE Trans. Antennas Propagat. 47 (1999) 1244–1253.
[20] K. Hill, C. Macon, T. Van, B. Welsh, P. Beyerle, A. MacFarland, E. Thiele, EMCC antenna benchmarks – Part I, Air Force Research Laboratory and ATK-Mission Research (2005).
[21] Z. Lou, J.M. Jin, Modeling and simulation of broad-band antennas using the time-domain finite element method, IEEE Trans. Antennas Propagat. 53 (2005) 4099–4110.
[22] T. Ozdemir, M.W. Nurnberger, J.L. Volakis, R. Kipp, J. Berrie, A hybridization of finite element and high-frequency methods for pattern prediction for antennas on aircraft structures, IEEE Antennas Propagat. Mag. 38 (1996) 28–38.
[23] J. Liu, J.M. Jin, Analysis of conformal antennas on a complex platform, Microwave Opt. Tech. Lett. 36 (2003) 139–142.
[24] Z. Lou, J.M. Jin, Finite element analysis of phased array antennas, Microwave Opt. Tech. Lett. 40 (2004) 490–496.