Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 222 (2007) 408–427

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A dual-field domain-decomposition method for the time-domain finite-element analysis of large finite arrays

Zheng Lou, 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

Received 13 March 2006; received in revised form 21 July 2006; accepted 26 July 2006
Available online 11 September 2006

Abstract

A novel dual-field time-domain finite-element domain-decomposition method is presented for an efficient and broadband numerical simulation of electromagnetic properties of large finite arrays. Instead of treating the entire array as a single computation domain, the method considers each array element as a smaller subdomain and computes both the electric and magnetic fields inside each subdomain. Adjacent subdomains are related to each other by the equivalent surface currents on the subdomain interfaces in an explicit manner. Furthermore, the method exploits the identical geometry of the array elements and further reduces the memory requirement and CPU time. The proposed method is highly efficient for the simulation of large finite arrays. Numerical stability and computational performance of the method are discussed. Several radiation examples are presented to demonstrate the accuracy and efficiency of the method.

Keywords: Finite-element method; Time-domain simulation; Domain decomposition; Finite arrays

1. Introduction

Large array structures have many scientific and engineering applications. For example, large antenna arrays are often formed to simulate a large effective antenna aperture, which results in a sharpened beam pattern and improved scan performance. In remote sensing, these large antenna arrays are used to achieve higher image resolutions. As another example, the photonic bandgap (PBG) structures have inspired great interests and found many applications in optics. Periodic dielectric structures are used to form resonant cavities and waveguides for optical light. Numerical simulation of large arrays has been considered challenging because of the tremendous computational resources it demands. Simulation of very large arrays can be made relatively simple by assuming the array is infinitely periodic. Under this assumption, the numerical analysis can be confined to a single unit cell by imposing appropriate periodic boundary condition and periodic radiation boundary condition. Such boundary conditions have been developed in both the frequency domain [1,2] and time

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

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.07.024

domain [3,4]. On the other hand, a faithful modeling of the finite array structure, although considered challenging, is still necessary in many applications. Such a modeling using the conventional methods often results in linear systems of millions of unknowns, requiring huge amount of memory and floating-point operations. In the attempts to model large finite arrays more efficiently, various fast algorithms have been developed. Among these algorithms, the most notable ones are the array decomposition method [5], the fast spectral domain algorithm [6], the characteristic basis function method [7], and the domain decomposition methods [8,9].

Recent developments in the time-domain finite-element method (TDFEM) have made it a powerful and versatile numerical technique for simulating a variety of complicated electromagnetic problems [10–18]. A conventional TDFEM solves the second-order vector wave equations using Galerkin's method [11]. Then a Newmark-beta method [19] is usually applied to obtain an unconditionally stable time-marching system. This method is also referred to as the implicit TDFEM, since a linear system of equations needs to be solved at each time step. As expected, this method is typically associated with relatively high computational complexity, which has significantly limited its capability of solving large-scale electromagnetic problems, especially the simulation of large finite arrays. In order to minimize the total CPU time, solving the linear system using a direct solver is preferable since the factorization can be reused at each time step. For large problems, however, the memory storage required by a direct solver becomes impractically large, even for the factorization of a sparse matrix resulting from an FEM discretization. As a result, an implicit TDFEM often resorts to an iterative solver for large problems to avoid excessive memory usage. However, the convergence of an iterative solver heavily depends on the properties of the matrix and the preconditioner it employs. An alternative approach to solving large-scale problems is to divide the original computation domain into several smaller subdomains. With a reduced size, each smaller subdomain problem can then be factorized and solved using a sparse direct solver and the overall computational complexity can be reduced as compared to the original single-domain problem. To further reduce the computation time, the subdomain problems can be distributed on a massively parallel computing system and solved in parallel. Therefore, an efficient domain-decomposition method together with the TDFEM provides a promising solution to the modeling of large finite arrays.

Recently, a domain-decomposition method referred to as the finite-element tearing and interconnecting (FETI) method has been developed [20,21]. It uses Lagrange multipliers to enforce the interface continuity condition and to formulate a reduced-order global problem. The method was recently extended to solve Maxwell's equations in the frequency domain [22,23] and time domain [24]. In the FETI method, the global problem is usually solved by an iterative solver, whose convergence depends on the shape and size of the subdomains and the total number of subdomains [21,25]. Furthermore, in a time-domain simulation, the local matrices for subdomain problems need to be factorized only once at the beginning of the computation, while the global problem has to be solved repeatedly using an iterative solver at each time step. As a result, the solution of the global problem can easily become the bottleneck of the entire computation. More recently, a novel domain-decomposition scheme that does not require solving a global problem has been proposed [26]. The method, which has been referred to as the dual-field domain-decomposition (DFDD) TDFEM, solves the dual-field second-order vector wave equations in each subdomain and relates the adjacent subdomains explicitly using the equivalent surface currents on the subdomain interfaces. Since adjacent subdomains are explicitly related at each time step, no global interface problem needs to be formulated and solved.

In this paper, we propose using the DFDD–TDFEM for the efficient modeling of large finite arrays. In an array problem, each array element is conveniently considered as a subdomain. The size of a single array element is usually small enough such that each subdoamin FEM system is small enough to be solved by a sparse direct solver. By doing so, the system matrix for each subdomain can be pre-factorized and stored in the memory before time marching. At each time step, each subdomain problem is solved efficiently using its local prefactorized matrix. The method further exploits the fact that most array elements have identical geometries and consequently identical subdomain matrices need to be factorized only once. This leads to a significant reduction in the memory requirement and factorization CPU time when the array size is large.

In the following sections, we first present the formulation of the DFDD–TDFEM. Next, a specialized DFDD array solver for the modeling of large finite arrays is described, and is followed by a numerical stability analysis. Finally, several numerical examples are presented and the computational performance of the method is compared with the conventional TDFEM.

2. Dual-field domain decomposition

The formulation of the DFDD–TDFEM is derived in this section. A computation domain consisting of two subdomains is considered first for the illustrative purpose. Arbitrary partitioning of the computation domain can be handled in the same fashion.

Fig. 1 shows a general computation domain V bounded by a metallic surface S_{M} and an impedance surface S_{A}. An arbitrary artificial boundary S_{B} breaks V into two subdomains V_1 and V_2. The electric and magnetic fields in subdomain V_m are denoted respectively as $\mathbf{E}_m$ and $\mathbf{H}_m$ where $m = 1, 2$. It can be shown that $\mathbf{E}_1$ and $\mathbf{H}_2$ satisfy the following second-order wave equations:

$$\nabla \times \left(\frac{1}{\mu_r}\nabla \times \mathbf{E}_1\right) + \frac{1}{c_0^2}\epsilon_r \frac{\partial^2 \mathbf{E}_1}{\partial t^2} + \mu_0 \sigma \frac{\partial \mathbf{E}_1}{\partial t} = -\mu_0 \frac{\partial \mathbf{J}_{i1}}{\partial t} \tag{1}$$

$$\nabla \times \left(\frac{1}{\epsilon_r}\nabla \times \mathbf{H}_2\right) + \frac{1}{c_0^2}\mu_r \frac{\partial^2 \mathbf{H}_2}{\partial t^2} + \frac{\mu}{\epsilon_r}\sigma \frac{\partial \mathbf{H}_2}{\partial t} = \nabla \times \left(\frac{1}{\epsilon_r}\mathbf{J}_{i2}\right) \tag{2}$$

where $\mathbf{J}_{i1}$ and $\mathbf{J}_{i2}$ denote the interior excitation in V_1 and V_2, respectively. The boundary conditions for the metallic surface S_{M} are

$$\hat{n} \times \mathbf{E} = 0 \tag{3}$$

$$\hat{n} \times (\nabla \times \mathbf{H}) = 0 \tag{4}$$

and the boundary conditions for the impedance surface S_{A} are

$$\hat{n} \times \left(\frac{1}{\mu}\nabla \times \mathbf{E}\right) + \gamma_e \frac{\partial}{\partial t}(\hat{n} \times \hat{n} \times \mathbf{E}) = 0 \tag{5}$$

$$\hat{n} \times \left(\frac{1}{\epsilon}\nabla \times \mathbf{H}\right) + \gamma_h \frac{\partial}{\partial t}(\hat{n} \times \hat{n} \times \mathbf{H}) = 0 \tag{6}$$

where γ_e and γ_h are related to the surface impedance. Testing (1) and (2) with a vector basis function $\mathbf{N}_i$ and integrating over V_1 and V_2 respectively yields the weak-form wave equations

$$\begin{aligned}&\int\int\int_{V_1}\left[\frac{1}{\mu_r}(\nabla \times \mathbf{N}_i)\cdot(\nabla \times \mathbf{E}_1) + \frac{\epsilon_r}{c_0^2}\mathbf{N}_i \cdot \frac{\partial^2 \mathbf{E}_1}{\partial t^2} + \mu_0 \sigma \mathbf{N}_i \cdot \frac{\partial \mathbf{E}_1}{\partial t}\right]\mathrm{d}V \\ &\quad = -\mu_0 \int\int\int_{V_1} \mathbf{N}_i \cdot \frac{\partial \mathbf{J}_{i1}}{\partial t}\mathrm{d}V + \int\int_{S_{A1}+S_{\mathrm{B}}} \hat{n}\cdot\left(\mathbf{N}_i \times \frac{1}{\mu_r}\nabla \times \mathbf{E}_1\right)\mathrm{d}S\end{aligned} \tag{7}$$

$$\begin{aligned}&\int\int\int_{V_2}\left[\frac{1}{\epsilon_r}(\nabla \times \mathbf{N}_i)\cdot(\nabla \times \mathbf{H}_2) + \frac{\mu_r}{c_0^2}\mathbf{N}_i \cdot \frac{\partial^2 \mathbf{H}_2}{\partial t^2} + \frac{\mu}{\epsilon_r}\sigma \mathbf{N}_i \cdot \frac{\partial \mathbf{H}_2}{\partial t}\right]\mathrm{d}V \\ &\quad = \int\int\int_{V_2} \frac{1}{\epsilon_r}(\nabla \times \mathbf{N}_i)\cdot \mathbf{J}_{i2}\mathrm{d}V + \int\int_{S_{A2}+S_{\mathrm{B}}} \hat{n}\cdot\left(\mathbf{N}_i \times \frac{1}{\epsilon_r}\nabla \times \mathbf{H}_2\right)\mathrm{d}S.\end{aligned} \tag{8}$$

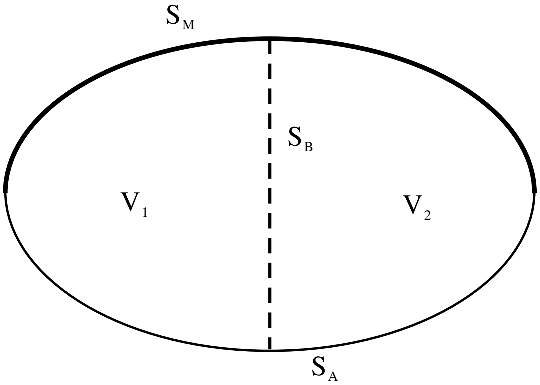

Fig. 1. Domain decomposition with two subdomains.

On S_A, we impose the first-order absorbing boundary conditions (ABC) by enforcing (5) and (6) and setting $\gamma_e = Y = \sqrt{\epsilon/\mu}$ and $\gamma_h = Z = \sqrt{\mu/\epsilon}$. On the interface S_B, $\mathbf{E}_1$ and $\mathbf{H}_2$ are, according to Maxwell's equations and the field continuity conditions, related by

$$\hat{n} \times \frac{1}{\mu_r}\nabla \times \mathbf{E}_1 = -\mu_0 \hat{n} \times \frac{\partial \mathbf{H}_2}{\partial t} \quad \text{on } S_B \tag{9}$$

$$\hat{n} \times \frac{1}{\epsilon_r}\nabla \times \mathbf{H}_2 = \epsilon_0 \hat{n} \times \frac{\partial \mathbf{E}_1}{\partial t} + \frac{\sigma}{\epsilon_r}\hat{n} \times \mathbf{E}_1 \quad \text{on } S_B. \tag{10}$$

For simplicity, we introduce the equivalent surface currents

$$\mathbf{J}_s = \hat{n} \times \mathbf{H}_2 \quad \text{on } S_B \tag{11}$$

$$\mathbf{M}_s = -\hat{n} \times \mathbf{E}_1 \quad \text{on} \quad S_B. \tag{12}$$

Using the notations for the equivalent surface currents, (7) and (8) can be rewritten as

$$\begin{aligned}&\iiint_{V_1} \left[\frac{1}{\mu_r}(\nabla \times \mathbf{N}_i)\cdot(\nabla \times \mathbf{E}_1) + \frac{\epsilon_r}{c_0^2}\mathbf{N}_i \cdot \frac{\partial^2 \mathbf{E}_1}{\partial t^2} + \mu_0 \sigma \mathbf{N}_i \cdot \frac{\partial \mathbf{E}_1}{\partial t}\right]\mathrm{d}V \\ &\quad + \mu_0 \iint_{S_{A1}} Y(\hat{n}\times\mathbf{N}_i)\cdot\left(\hat{n}\times\frac{\partial \mathbf{E}_1}{\partial t}\right)\mathrm{d}S = -\mu_0 \iiint_{V_1} \mathbf{N}_i \cdot \frac{\partial \mathbf{J}_{i1}}{\partial t}\mathrm{d}V \\ &\quad + \mu_0 \iint_{S_B} (\hat{n}\times\mathbf{N}_i)\cdot\left[\hat{n}\times\frac{\partial \mathbf{J}_s}{\partial t}\right]\mathrm{d}S\end{aligned} \tag{13}$$

$$\begin{aligned}&\iiint_{V_2} \left[\frac{1}{\epsilon_r}(\nabla \times \mathbf{N}_i)\cdot(\nabla \times \mathbf{H}_2) + \frac{\mu_r}{c_0^2}\mathbf{N}_i \cdot \frac{\partial^2 \mathbf{H}_2}{\partial t^2} + \frac{\mu}{\epsilon_r}\sigma \mathbf{N}_i \cdot \frac{\partial \mathbf{H}_2}{\partial t}\right]\mathrm{d}V \\ &\quad + \epsilon_0 \iint_{S_{A2}} Z(\hat{n}\times\mathbf{N}_i)\cdot\left(\hat{n}\times\frac{\partial \mathbf{H}_2}{\partial t}\right)\mathrm{d}S = \iiint_{V_2} \frac{1}{\epsilon_r}(\nabla\times\mathbf{N}_i)\cdot\mathbf{J}_{i2}\mathrm{d}V \\ &\quad + \epsilon_0 \iint_{S_B} (\hat{n}\times\mathbf{N}_i)\cdot\left[\hat{n}\times\frac{\partial \mathbf{M}_s}{\partial t}\right]\mathrm{d}S + \iint_{S_B}\frac{\sigma}{\epsilon_r}(\hat{n}\times\mathbf{N}_i)\cdot(\hat{n}\times\mathbf{M}_s)\,\mathrm{d}S.\end{aligned} \tag{14}$$

Next, both $\mathbf{E}_1$ and $\mathbf{H}_2$ are expanded using the vector basis function $\mathbf{N}_i$. After standard spatial discretization procedures, (13) and (14) can be written in the following matrix form:

$$[S_e]\{e_1\} + \frac{1}{c_0^2}[M_e]\frac{\partial^2\{e_1\}}{\partial t^2} + \frac{1}{c_0}[B_e]\frac{\partial\{e_1\}}{\partial t} + \frac{1}{c_0}[A_e]\frac{\partial\{e_1\}}{\partial t} = -\frac{1}{c_0}\frac{\partial\{f\}}{\partial t} + \frac{1}{c_0}\frac{\partial\{j\}}{\partial t} \tag{15}$$

$$[S_h]\{h_2\} + \frac{1}{c_0^2}[M_h]\frac{\partial^2\{h_2\}}{\partial t^2} + \frac{1}{c_0}[B_h]\frac{\partial\{h_2\}}{\partial t} + \frac{1}{c_0}[A_h]\frac{\partial\{h_2\}}{\partial t} = \{g\} + \frac{1}{c_0}\frac{\partial\{m\}}{\partial t} + \{\tilde{m}\} \tag{16}$$

where the matrix entries are given by

$$S_e(i,j) = \iiint_{V_1} \frac{1}{\mu_r}(\nabla\times\mathbf{N}_i)\cdot(\nabla\times\mathbf{N}_j)\mathrm{d}V \tag{17}$$

$$M_e(i,j) = \iiint_{V_1} \epsilon_r \mathbf{N}_i \cdot \mathbf{N}_j \,\mathrm{d}V \tag{18}$$

$$B_e(i,j) = Z_0 \iiint_{V_1} \sigma \mathbf{N}_i \cdot \mathbf{N}_j \,\mathrm{d}V \tag{19}$$

$$A_e(i,j) = Z_0 \iint_{S_{A1}} Y(\hat{n}\times\mathbf{N}_i)\cdot(\hat{n}\times\mathbf{N}_j)\,\mathrm{d}S \tag{20}$$

and

$$S_h(i,j) = \iiint_{V_2} \frac{1}{\epsilon_r}(\nabla\times\mathbf{N}_i)\cdot(\nabla\times\mathbf{N}_j)\,\mathrm{d}V \tag{21}$$

$$M_h(i,j) = \iiint_{V_2} \mu_r \mathbf{N}_i \cdot \mathbf{N}_j \,\mathrm{d}V \tag{22}$$

$$B_h(i,j) = Z_0 \int\int\int_{V_2} \frac{\mu_r}{\epsilon_r} \sigma \mathbf{N}_i \cdot \mathbf{N}_j \, \mathrm{d}V \tag{23}$$

$$A_h(i,j) = Y_0 \int\int_{S_{A2}} Z(\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{N}_j) \, \mathrm{d}S \tag{24}$$

where $Z_0 = \sqrt{\mu_0/\epsilon_0}$, $Y_0 = \sqrt{\epsilon_0/\mu_0}$ and the right-hand side vector entries are given by

$$f(i) = Z_0 \int\int\int_{V_1} \mathbf{N}_i \cdot \mathbf{J}_{i1} \, \mathrm{d}V \tag{25}$$

$$g(i) = \int\int\int_{V_2} \frac{1}{\epsilon_r} (\nabla \times \mathbf{N}_i) \cdot \mathbf{J}_{i2} \, \mathrm{d}V \tag{26}$$

$$j(i) = Z_0 \int\int_{S_B} (\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{J}_s) \, \mathrm{d}S \tag{27}$$

$$m(i) = Y_0 \int\int_{S_B} (\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{M}_s) \, \mathrm{d}S \tag{28}$$

$$\tilde{m}(i) = \int\int_{S_B} \frac{\sigma}{\epsilon_r} (\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{M}_s) \, \mathrm{d}S. \tag{29}$$

Note that the electric field in subdomain V_1 and the magnetic field in subdomain V_2 are related to each other through the equivalent surface currents on the interface. Therefore, (15) and (16) form a pair of coupled equations that can be used to solve for the fields in both subdomains.

To construct a valid time-marching scheme, we sample the electric field on integer time indices $n = 0$, 1, $2, \ldots, N$ and the magnetic field on half integer time indices $n = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots, \frac{2N+1}{2}$, similar to the finite-difference time-domain (FDTD) method. Inside each subdomain, however, the Newmark-beta method is used for temporal discretization. This results in the following time-marching system:

$$\begin{aligned} & c_0^2 \Delta t^2 [S_e](\beta\{e_1^{n-1}\} + (1-2\beta)\{e_1^n\} + \beta\{e_1^{n+1}\})[M_e](\{e_1^{n-1}\} - 2\{e_1^n\} + \{e_1^{n+1}\}) \\ & \quad + \frac{c_0 \Delta t}{2} (B_e][\{e_1^{n+1}\} - \{e_1^{n-1}\}) + \frac{c_0 \Delta t}{2} [A_e][\{e_1^{n+1}\} - \{e_1^{n-1}\}] \\ & = -c_0 \Delta t \left(\{f^{n+\frac{1}{2}}\} - \{f^{n-\frac{1}{2}}\} \right) + c_0 \Delta t \left(\{j^{n+\frac{1}{2}}\} - \{j^{n-\frac{1}{2}}\} \right) \end{aligned} \tag{30}$$

$$\begin{aligned} & c_0^2 \Delta t^2 [S_h] \left(\beta\{h_2^{n-\frac{1}{2}}\} + (1-2\beta)\{h_2^{n+\frac{1}{2}}\} + \beta\{h_2^{n+\frac{3}{2}}\} \right) [M_h] \left(\{h_2^{n-\frac{1}{2}}\} - 2\{h_2^{n+\frac{1}{2}}\} + \{h_2^{n+\frac{3}{2}}\} \right) \\ & \quad + \frac{c_0 \Delta t}{2} (B_h] \left(\{h_2^{n+\frac{3}{2}}\} - \{h_2^{n-\frac{1}{2}}\} \right) + \frac{c_0 \Delta t}{2} [A_h] \left(\{h_2^{n+\frac{3}{2}}\} - \{h_2^{n-\frac{1}{2}}\} \right) \\ & = c_0^2 \Delta t^2 \{g^n\} + c_0 \Delta t (\{m^{n+1}\} - \{m^n\}) + \frac{1}{2} c_0^2 \Delta t^2 (\{\tilde{m}^{n+1}\} + \{\tilde{m}^n\}). \end{aligned} \tag{31}$$

According to (30) and (31), each time step in the time marching consists of two substeps. In the first substep, the electric field at time index $n+1$ is updated using the electric field at previous time indices (n and $n-1$) and the electric current on the interface at previous time indices ($n+\frac{1}{2}$ and $n-\frac{1}{2}$). In the second substep, the magnetic field at time index $n+\frac{3}{2}$ is updated using the magnetic field at previous time indices ($n+\frac{1}{2}$ and $n-\frac{1}{2}$) and the magnetic current on the interface at previous time indices ($n+1$ and n). At the end of the second substep, the time indices are incremented by one for all the quantities in (30) and (31) and the same procedures are repeated at the next time step.

The above time-marching system is derived from the coupled equations for $\mathbf{E}_1$ and $\mathbf{H}_2$. Note that a similar pair of coupled equations relating $\mathbf{E}_2$ and $\mathbf{H}_1$ also exists. Either one of the them can be used to solve for the primary fields in the two subdomains and then the dual fields can be obtained from Maxwell's equations. However, this is only applicable to the two-subdomain case. For a general multiple-subdomain scenario, it is necessary to solve for both the electric and magnetic fields in each subdomain. Suppose that the computation domain is divided into M non-overlapping subdomains V_m ($m = 1, 2, \ldots, M$). Then the electric field $\mathbf{E}_m$ and magnetic field $\mathbf{H}_m$ in subdomain V_m satisfy

$$\nabla \times \left(\frac{1}{\mu_r}\nabla \times \mathbf{E}_m\right) + \frac{1}{c_0^2}\epsilon_r \frac{\partial^2 \mathbf{E}_m}{\partial t^2} + \mu_0 \sigma \frac{\partial \mathbf{E}_m}{\partial t} = -\mu_0 \frac{\partial \mathbf{J}_{im}}{\partial t} \tag{32}$$

$$\nabla \times \left(\frac{1}{\epsilon_r}\nabla \times \mathbf{H}_m\right) + \frac{1}{c_0^2}\mu_r \frac{\partial^2 \mathbf{H}_m}{\partial t^2} + \frac{\mu}{\epsilon_r} \sigma \frac{\partial \mathbf{H}_m}{\partial t} = \nabla \times \left(\frac{1}{\epsilon_r}\mathbf{J}_{im}\right). \tag{33}$$

The same discretization scheme as described earlier can be employed, which leads to matrix equations similar to (15) and (16). Assuming that V_m has L_m neighbors and thus L_m interfaces, then (27)–(29) need to be modified as:

$$\{j\} = Z_0 \sum_{l=1}^{L_m} \int\int_{S_{\mathrm{B}l}} (\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{J}_s)\,\mathrm{d}S \tag{34}$$

$$\{m\} = Y_0 \sum_{l=1}^{L_m} \int\int_{S_{\mathrm{B}l}} (\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{M}_s)\,\mathrm{d}S \tag{35}$$

$$\{\tilde{m}\} = \sum_{l=1}^{L_m} \int\int_{S_{\mathrm{B}l}} \frac{\sigma}{\epsilon_r} (\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{M}_s)\,\mathrm{d}S \tag{36}$$

where

$$\mathbf{J}_s = \hat{n} \times \mathbf{H}_{m_l} \quad \text{on } S_{\mathrm{B}} \tag{37}$$

$$\mathbf{M}_s = -\hat{n} \times \mathbf{E}_{m_l} \quad \text{on } S_{\mathrm{B}} \tag{38}$$

where m_l denotes the subdomain that is connected to subdomain m through interface l. Note that in (34)–(36) the equivalent surface currents are always evaluated using the electromagnetic fields in the neighboring subdomains, as similar to the two-subdomain case.

3. DFDD array solver

In the array configurations where all the array elements have identical geometries, the entire array can be characterized sufficiently by a single element, which is typically referred to as the unit cell. The repetition of the geometry can be exploited in the numerical simulation and a better efficiency can be achieved. In the specialized DFDD array algorithm, each array element is considered as an individual subdomain and identical spatial discretizations are applied to all of them except for the array elements sitting on the edges and corners of the array. As a result, the FEM system matrices for all subdomains are the same except for the corner and edge elements. For identical subdomains, the system matrix only has to be generated and factorized once. For large arrays, this algorithm can lead to a significant reduction in the total memory usage and factorization time.

Fig. 2 shows the partition of a typical two-dimensional array into a central region, four edge regions, and four corner regions. Due to the enforcement of exterior boundary conditions such as the ABC and the perfectly matched layers (PML), the elements in the edge regions and corner regions are in general different from those in the central region. Inside each region, however, all the array elements are identical to one another. Therefore, the entire array can be represented sufficiently by nine unit cells, which are denoted as unit cells I–IX in Fig. 2. Note that the separate treatment of the corner and edge elements is actually an advantage, rather than a disadvantage because it provides a great flexibility for the engineering design of finite arrays.

Since all the array elements are replicas of the nine unit cells, the FEM mesh only needs to be created for the nine unit cells. In order to enforce the field continuity on the interface, however, surface meshes on the opposite sides of the unit cell have to be identical, although this requirement can be relaxed with a modification of the algorithm. For each of the nine unit cells (subdomains), an FEM matrix is assembled, factorized, and stored in memory in the preprocessing stage. During the time marching, each of the array elements is updated using the appropriate factorized matrix according to its position in the array. Thus, this algorithm requires to store only nine factorized matrices regardless of the actual size of the array. Considering that matrix factorization typically dominates the memory consumption, a significant reduction in the memory requirement can be achieved by using this specialized array solver, especially for large arrays. The same argument can be made for the factorization time. The solving time, however, remains unaffected. Although the system matrices are

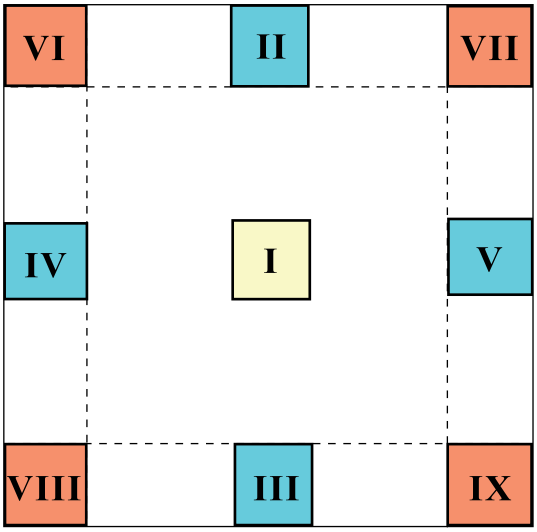

Fig. 2. Grouping of array elements into nine categories.

the same for subdomains with identical geometries, the unknown coefficients for these subdomains are all different. Consequently, the solving time is not reduced by the array algorithm.

4. Stability analysis

It has been shown that an implicit TDFEM system employing the Newmark-beta method with $\beta \geqslant 1/4$ is unconditionally stable [11]. This means that the time step can be chosen independent of the spatial discretization. In the DFDD–TDFEM, the Newmark-beta method is invoked inside each subdomain. On the subdomain interfaces, however, an explicit updating scheme is employed. Consequently, the entire scheme becomes conditionally stable. To derive the stability condition for the DFDD–TDFEM, we again consider the simplest case where the computation domain contains only two subdomains V_1 and V_2. Further, we set $\beta = 1/4$ in (30) and (31) in our stability analysis and ignore the lossy and excitation terms in (30) and (31), since these terms will not affect the stability criteria. After these simplifications, (30) and (31) become

$$\begin{aligned} & c_0^2\Delta t^2[S_e]\left(\frac{1}{4}\{e_1^{n-1}\}+\frac{1}{2}\{e_1^{n}\}+\frac{1}{4}\{e_1^{n+1}\}\right)+[M_e]\left(\{e_1^{n-1}\}-2\{e_1^{n}\}+\{e_1^{n+1}\}\right) \\ & \quad = c_0\Delta t Z_0[P]\left(\{h_2^{n+\frac{1}{2}}\}-\{h_2^{n-\frac{1}{2}}\}\right) \end{aligned} \tag{39}$$

$$\begin{aligned} & c_0^2\Delta t^2[S_h]\left(\frac{1}{4}\{h_2^{n-\frac{1}{2}}\}+\frac{1}{2}\{h_2^{n+\frac{1}{2}}\}+\frac{1}{4}\{h_2^{n+\frac{3}{2}}\}\right)+[M_h]\left(\{h_2^{n-\frac{1}{2}}\}-2\{h_2^{n+\frac{1}{2}}\}+\{h_2^{n+\frac{3}{2}}\}\right) \\ & \quad = c_0\Delta t Y_0[Q]\left(\{e_1^{n+1}\}-\{e_1^{n}\}\right) \end{aligned} \tag{40}$$

where matrices $[P]$ and $[Q]$ are given by

$$P(i,j) = \int\int_{S_\mathrm{B}} (\hat{n}\times\mathbf{N}_i)\cdot(\hat{n}\times\hat{n}\times\mathbf{N}_j)\,\mathrm{d}S \tag{41}$$

$$Q(i,j) = -\int\int_{S_\mathrm{B}} (\hat{n}\times\mathbf{N}_j)\cdot(\hat{n}\times\hat{n}\times\mathbf{N}_i)\,\mathrm{d}S. \tag{42}$$

Following the standard approach, we assume a plane-wave solution

$$\mathbf{E}_1(n,\mathbf{r}) = \mathbf{E}_1(\mathbf{r})\mathrm{e}^{\mathrm{j}\omega n\Delta t} \tag{43}$$

$$\mathbf{H}_2(n,\mathbf{r}) = \mathbf{H}_2(\mathbf{r})\mathrm{e}^{\mathrm{j}\omega n\Delta t}. \tag{44}$$

On substituting in (43) and (44), (39) and (40) are reduced to

$$\left(c_0^2\Delta t^2[S_e]\cos^2\frac{\omega\Delta t}{2}-4[M_e]\sin^2\frac{\omega\Delta t}{2}\right)\{e_1\}=2jc_0\Delta tZ_0[P]\sin\frac{\omega\Delta t}{2}\{h_2\} \tag{45}$$

$$\left(c_0^2\Delta t^2[S_h]\cos^2\frac{\omega\Delta t}{2}-4[M_h]\sin^2\frac{\omega\Delta t}{2}\right)\{h_2\}=2jc_0\Delta tY_0[Q]\sin\frac{\omega\Delta t}{2}\{e_1\}. \tag{46}$$

Defining a combined field vector $\{v\}=[\{e_1\}/\sqrt{Z_0},\sqrt{Z_0}\{h_2\}]^{\mathrm{T}}$ and combined matrices

$$[S]=\begin{bmatrix}S_e & 0\\ 0 & S_h\end{bmatrix},\quad [M]=\begin{bmatrix}M_e & 0\\ 0 & M_h\end{bmatrix},\quad [T]=\begin{bmatrix}0 & P\\ Q & 0\end{bmatrix}, \tag{47}$$

we can write (45) and (46) in a compact format as

$$\frac{1}{4}c_0^2\Delta t^2\cot^2\frac{\omega\Delta t}{2}[K]\{v\}-\frac{j}{2}c_0\Delta t\csc\frac{\omega\Delta t}{2}[L]\{v\}=\{v\} \tag{48}$$

where $[K]=[M]^{-1}[S]$ and $[L]=[M]^{-1}[T]$. Denote the eigenvalues of $[K]$ and $[L]$ as λ_K and λ_L, respectively. It can be easily seen from (41) and (42) that $[P]$ and $[Q]$ satisfy $[P]=-[Q]^T$. Therefore, $[T]$ is an anti-symmetric matrix. Given that $[M]$ is a positive definite and symmetric matrix and $[S]$ is a symmetric matrix, it follows that λ_K has to be purely real and λ_L has to be purely imaginary. From (48), it can be shown that λ_K and λ_L satisfy

$$\frac{1}{4}c_0^2\Delta t^2\cot^2\frac{\omega\Delta t}{2}\lambda_K-\frac{j}{2}c_0\Delta t\csc\frac{\omega\Delta t}{2}\lambda_L=1. \tag{49}$$

In order for ω to be real for arbitrary λ_K and λ_L, (49) requires that

$$\frac{1}{4}c_0^2\Delta t^2\lambda_K<\infty \tag{50}$$

$$\frac{1}{2}c_0\Delta t|\lambda_L|<1. \tag{51}$$

Therefore, the stability condition is given by

$$\Delta t<\frac{2}{c_0}\frac{1}{|\lambda_L|_{\max}}=\frac{2}{c_0}\frac{1}{\rho([L])} \tag{52}$$

where $\rho(\cdot)$ stands for the spectral radius of $(\cdot)$. The above analysis indicates that the updating system represented by (39) and (40) is conditionally stable and the stability condition is determined by the spectral property of the matrix $[L]=[M]^{-1}[T]$. It is worth noting that both $[P]$ and $[Q]$ are highly sparse matrices whose entries are zero except for those associated with the degrees of freedom on the interface S_{B}. Thus, matrices $[T]$ and $[L]$ are also highly sparse matrices. By using the properties of the FEM matrices, it can be shown [27] that matrix $[L]$ depends only on the spatial discretization immediately next to the interface. For example, if a three-dimensional (3D) tetrahedral mesh is employed, the stability condition depends only on the tetrahedrons immediately connected to the interface. Note that such a stability condition is less stringent than that of the FDTD method or the explicit TDFEM where the time step is determined by the smallest geometry/discretization in the entire computation domain. In the array simulations, small geometrical features, such as the antenna feeds, are usually contained inside each subdomain and the spatial discretization on the subdomain interfaces is relatively sparse and well-behaved. In such cases, the stability condition of the DFDD–TDFEM will be much more relaxed than that of the FDTD method or the explicit TDFEM.

To validate the stability condition (52), a 1D wave propagation problem is simulated using the DFDD–TDFEM. The 1D computation domain is 0.32 m in length and divided into two subdomains. In the first simulation, we employ a uniform mesh with mesh size $h_0=0.01$ m throughout the entire computation domain. In the second simulation, we employ a non-uniform mesh with $h(x)=\alpha(x)h_0$ where α decreases linearly from 1.0 at the interface to 0.06 at the two ends of the computation domain. In both cases, linear basis functions are used on each cell. The spectral radius of matrix $[L]$ and the calculated stability threshold $\Delta t_{\max}$ are shown in Table 1. The stability condition for the uniform and non-uniform discretizations

Table 1
Comparison of the stability conditions for a uniform and non-uniform 1D mesh

Discretization	$\rho([L])$	$\Delta t_{\max} = \frac{2}{c_0\rho([L])}$	Numerical tests	
			Stable	Unstable
Uniform	346.4 m^{-1}	19.3 ps	$\Delta t = 19.0$ ps	$\Delta t = 20.0$ ps
Non-uniform	348.2 m^{-1}	19.3 ps	$\Delta t = 19.0$ ps	$\Delta t = 20.0$ ps

are found to be almost identical to each other since both of them have the same mesh size at the interface. It is also observed that in both cases the theoretical predictions are consistent with numerical simulation results.

For 3D problems, the spectral property of $[L]$ depends on both the shape and dimensions of the elements and the properties of the basis functions employed. For an equilateral tetrahedron with a side length of h, the calculated stability condition is $\Delta t < 0.3h/c$ when the lowest-order basis functions are used. Through proper scaling, the stability condition for an arbitrary interface mesh can be estimated. A similar estimation can be made when higher-order basis functions are used.

5. Numerical examples

In this section, we consider several radiation problems to demonstrate the accuracy and efficiency of the DFDD–TDFEM. In all the examples, the excitation is a modulated Gaussian pulse with a central frequency located at the center of the band of interest. The open free space is terminated using the first-order ABC for efficiency. For a better absorbing performance, the PML can be used. It can be shown that the PML, when considered as anisotropic material layers, can be incorporated into the DFDD–TDFEM formulation in a straightforward manner.

5.1. Monopole array

We first consider a finite monopole array radiating over an infinite ground plane. The unit cell configuration of the monopole is shown in Fig. 3a. The monopole is formed by extending the central conductor of the coaxial cable 10 cm above the ground plane. The coaxial cable has an inner radius of 1 cm and an outer radius of 2.3 cm. First, a 10 × 1 one-dimensional array, which is sketched in Fig. 3b, is considered. The separation between adjacent array elements is 10 cm. The coaxial ports are excited simultaneously with a zero phase progression. The incident pulse is a modulated Gaussian pulse with central frequency $f_0 = 750$ MHz. In the DFDD–TDFEM simulation, the entire computation domain is partitioned into 36 subdomains. Each subdomain is in turn discretized into approximately 4000 tetrahedron elements and hierarchical basis functions of mixed-second order [28] are used for field expansion. The resultant number of unknowns is approximately

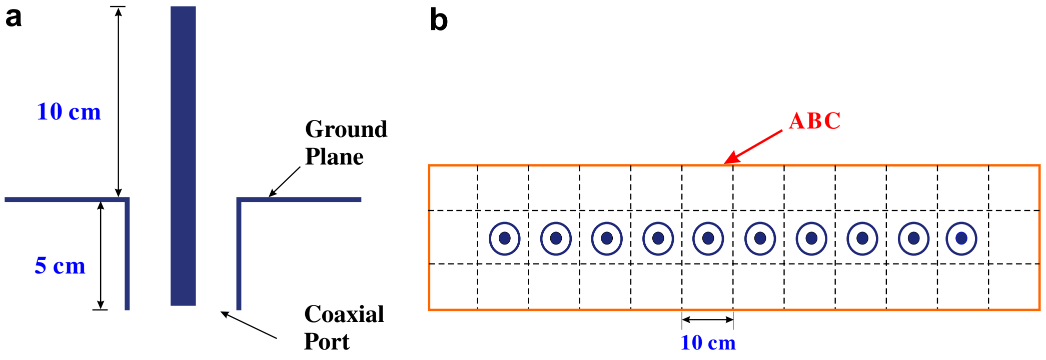

Fig. 3. (a) A Unit cell monopole antenna. (b) A 10 × 1 monopole array. Dashed lines denote the subdomain interfaces.

24,000 in each subdomain and about 800,000 for the entire problem. To demonstrate the accuracy of the method, the active reflection coefficients at two sample ports are calculated as a function of frequency. Port 1 is located at the left-most of the array while Port 5 is located near the center of the array. Fig. 4 shows the calculated active reflection coefficients at the two sample ports over the frequency range of 550–950 MHz. The DFDD–TDFEM results are compared with the results obtained by the frequency-domain FETI [9] for validation. Good agreement between the two numerical methods is found. Fig. 5 shows the magnitude variation of the reflection coefficients across the array from Port 1 to 10 at 750 MHz. It is clearly seen that a large magnitude variation appears at the two ends of the array due to the well-known edge effect of a finite array while the magnitude of the central elements remains relatively constant.

Next, we consider two larger square arrays consisting of the same monopoles. The array separation is 10 cm in both the x- and y-directions. The radiation patterns of a 9×9 and 30×30 monopole arrays at 750 MHz are shown in Fig. 6a and b, respectively. Again, the DFDD–TDFEM results are compared with the frequency-domain FETI results for validation. In both cases, the array is configured for an endfire radiation ($\theta = 90°$). A 90°- and 0°-phase progression is applied to the x- and y-directions, respectively. In the time-domain simulation, the phase progression is realized by applying an appropriate time delay to the excitation at different ports. For the 30×30 array, the total number of unknowns is approximately 33 million.

The monopole array examples are computed using a serial code on a SGI Altix 350 system with Intel-Itanium II 1.5 GHz processors. The entire computation consists of two stages. In the preprocessing stage, the system matrices for each of the nine unit cell subdomains are assembled and pre-factorized sequentially

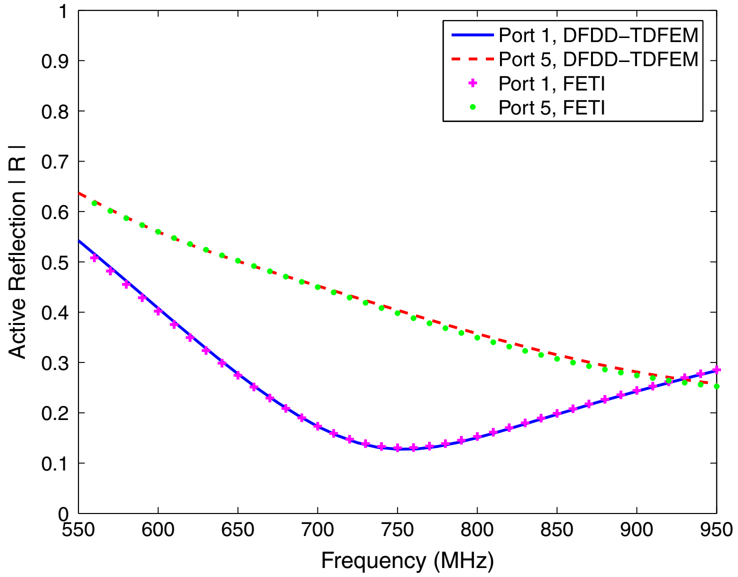

Fig. 4. Magnitude of the active reflection coefficients at Port 1 and Port 5.

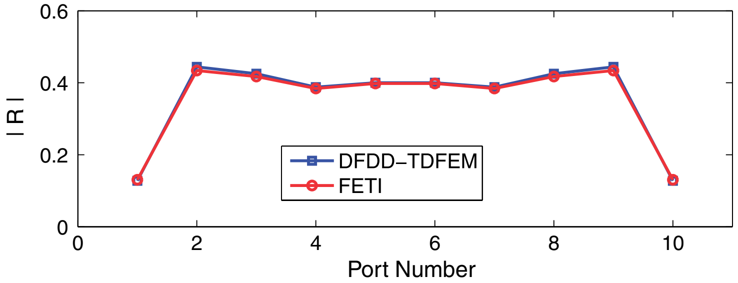

Fig. 5. Magnitude variation in the reflection coefficients across the array at 750 MHz.

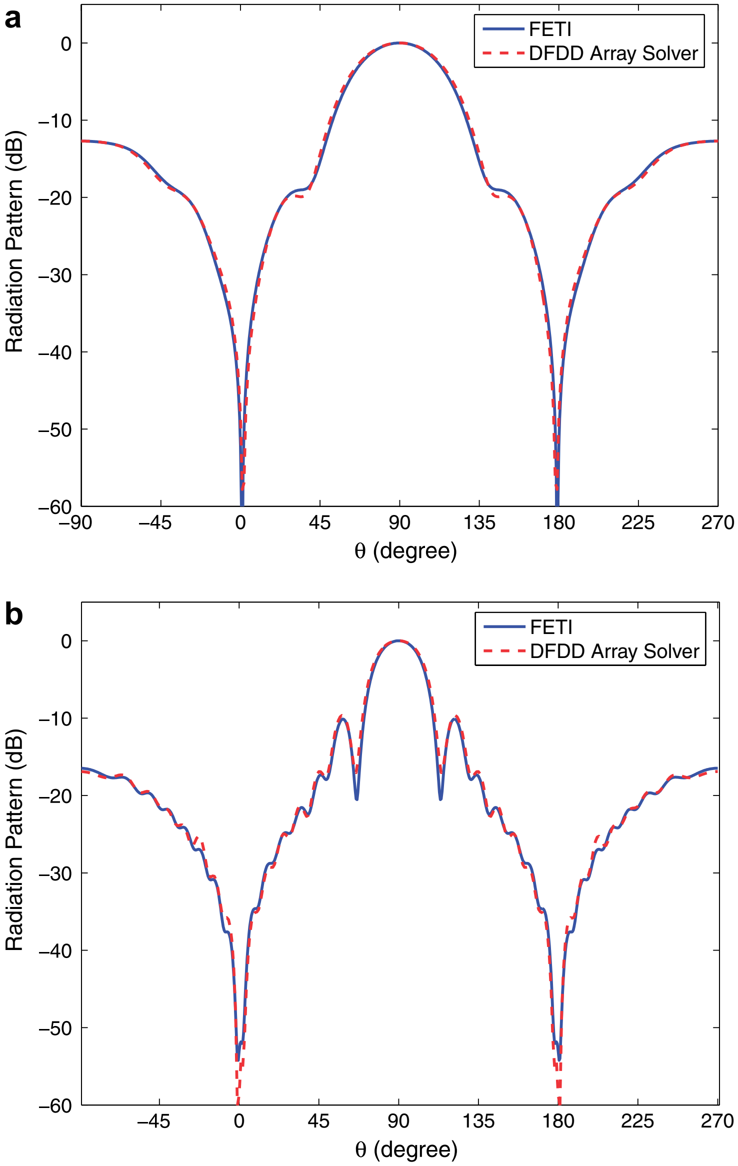

Fig. 6. Endfire pattern of the monopole array at 750 MHz. (a) 9 × 9 array and (b) 30 × 30 array.

and stored in the memory. In the time-marching stage, the fields in each subdomain are updated also in a sequential manner at each time step using the appropriate factorized matrix. Here we use the sparse direct solver provided in SGI's SCSL library for matrix factorizations. The memory requirement and the CPU time of the DFDD–TDFEM array solver are shown in Figs. 7–9. Since the matrix factorization is performed only once at the beginning, its CPU time is recorded separately. Fig. 7 shows the peak memory usage of the DFDD–TDFEM array solver versus array size. When the array size is increased from 1 × 1 to 9 × 9, the memory requirement of the DFDD–TDFEM array solver is increased only slightly from 681 to 837 MB. This is because the memory allocated for matrix factorization is independent of the array size. For even larger arrays (30 × 30 for example), the memory usage starts to increase with the array size. This is because the memory used by matrix factorization becomes only a small portion of the total memory when the array size becomes very large and the majority of the memory is allocated to store the unknown coefficients. However, such an increase is only linear with respect to the number of unknowns. For comparison, the memory required by a conven-

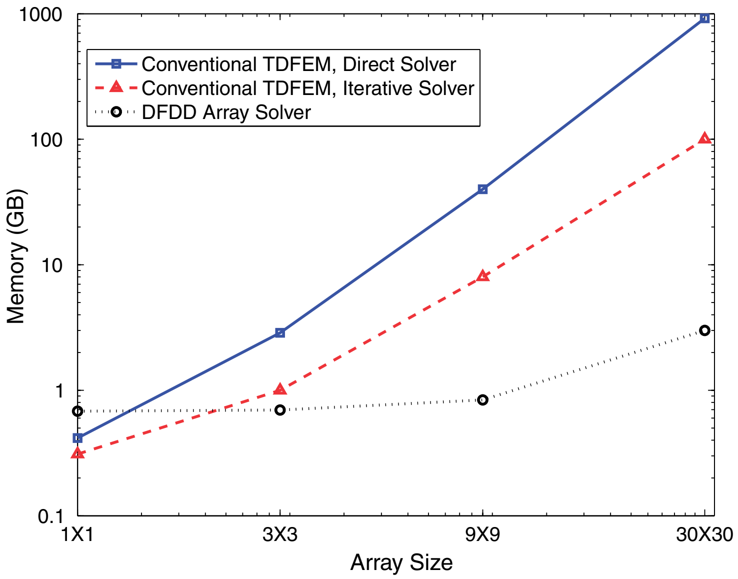

Fig. 7. Memory requirement versus array size for the monopole array example.

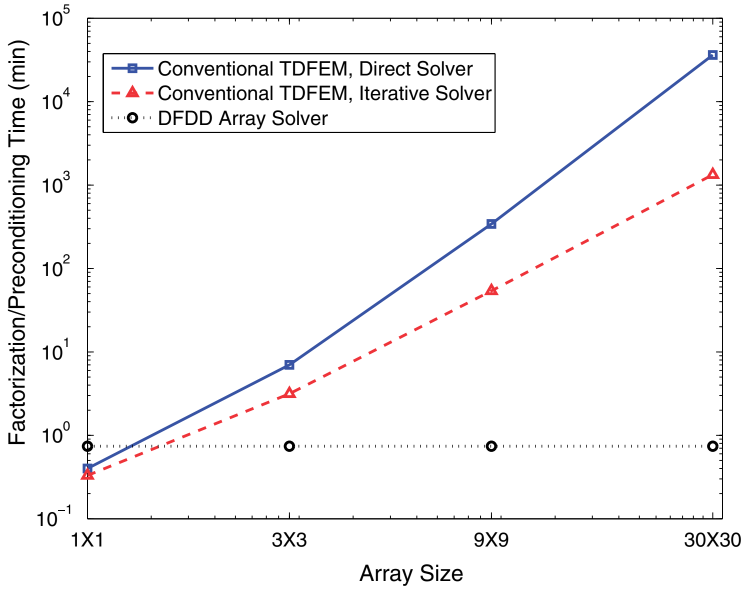

Fig. 8. Factorization/preconditioning CPU time versus array size for the monopole array example.

tional TDFEM is also plotted in Fig. 7 (note that the memory requirements for very large array sizes are projected data). Since the memory requirement depends on the linear solver, both the direct and iterative solvers are used in the conventional TDFEM. For the direct solver, we use SGI's SCSL library. For the iterative solver, we use the GMRES with an incomplete Cholesky preconditioner. For the 1×1 array, it is seen from the figure that the conventional TDFEM takes even less memory than the DFDD–TDFEM array solver. This is due to the fact that the DFDD–TDFEM solves for both the electric and magnetic fields while the conventional method calculates only the electric (or magnetic) field. Thus, the total number of unknowns in the DFDD–TDFEM solution is doubled. However, when the array size grows larger, the memory requirement for the conventional TDFEM increases quickly. For the 9×9 array, for instance, a total memory of 40 GB would be

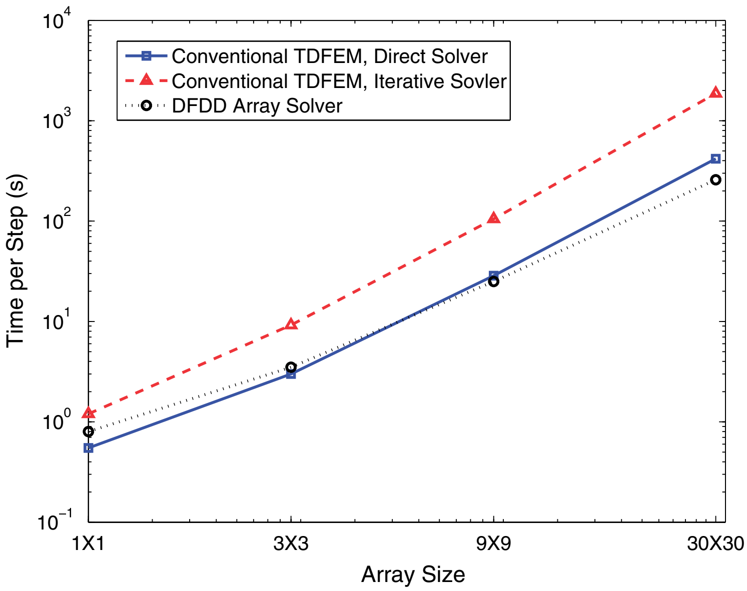

Fig. 9. Time-marching CPU time per step versus array size for the monopole array example.

required by the conventional TDFEM using a direct solver in contrast to the 837 MB of memory required by the DFDD–TDFEM array solver. Although the total memory requirement can be reduced by using an iterative solver instead of a direct solver, the memory increase with the problem size is still faster than a linear increase.

Fig. 8 shows the factorization CPU time versus array size for the DFDD–TDFEM array solver and the conventional TDFEM. It is expected that the factorization CPU time remains constant for the DFDD–TDFEM array solver while it shows an O(N^2) increase with respect to the number of unknowns for the conventional TDFEM using a direct solver. When an iterative solver is used in the conventional TDFEM, the preconditioning time is shown instead, which exhibits a slower increase rate than the direct factorization. Finally, the solving time per time step during the time marching is shown in Fig. 9. Also as expected, the computational complexity of the DFDD–TDFEM array solver is comparable to that of the conventional TDFEM using a direct solver, both of which are close to O(N). However, the solving time for the conventional TDFEM using an iterative solver is significantly longer than both of them because of the iterations. Note that the solving time for an iterative solver varies considerably with different types of preconditioners. Usually, a good preconditioner improves convergence but also requires a large amount of memory and longer perconditioning time.

The CPU time shown in Figs. 8 and 9 are recorded on a single processor. As we mentioned earlier, one of the major advantages of the proposed method is that it can be easily implemented as a communication-efficient parallel algorithm. In such a parallel algorithm, each processor is assigned one or several subdomain problems. Each processor then processes the subdomain problems according to the algorithm described previously. No global matrix needs to be assembled or solved. During the time marching, each processor first solves its local subdomain problems and then they are synchronized for communication, which is realized by exchanging the surface equivalent currents on the inter-processor interfaces. Compared to the solving of local matrices, the overhead for the inter-processor communication is usually negligible. Thus, a very high parallel efficiency can be achieved.

To demonstrate the efficiency of the parallel algorithm, we repeat the calculation of the 30 × 30 monopole array. Now the computation is distributed on a cluster of Apple Xserver 2.0 GHz G5 processors. Table 2 shows an almost linear decrease of solving time with an increasing number of processors. Since the inter-processor communication cost is negligible compared to the solving of local subdomain problems, a high parallel efficiency mainly depends on a good balance of work load among available processors.

Table 2
Time-marching CPU time per time step versus number of processors for the 30 × 30 monopole array

Number of processors	Solving time per step (s)
1	328
25	22
64	6.8
144	2.5

5.2. Vivaldi antenna array

The Vivaldi antenna is generally recognized as an example of ultra wide-band antennas. Although different configurations of the Vivaldi antennas exist, all of them contain a gradually flared notch structure to provide smooth impedance transition to free space. The unit cell configuration considered here was originally described in [2] and shown in Fig. 10. The model comprises a conducting patch printed on one side of the substrate, which stands vertically above a ground plane. A narrow slotline is cut in the middle of the conducting patch and is gradually flared into an open mouth at the interface to free space. The antenna is fed by a stripline from below the ground plane. A small section of coplanar waveguide (CPW) is inserted between the stripline feed and the slotline of the antenna. The coplanar waveguide consists of two slots, one of which evolves as the flared slotline and the other ends at a hollow circle cut on the conducting patch. The hollow circle serves as a wide-band open circuit that contributes to the broadband behavior [29]. The advantage of this design is that metallization is applied to only one side of the substrate and no via-holes are needed for ground connections. The various dimensions and parameters of the antenna are given in Fig. 10.

Here we consider a 10 × 10 Vivaldi array which is shown in Fig. 11. The spacing between array elements is 40 mm in both x- and y-directions. The separation corresponds to one half of a wavelength at 3.75 GHz. An infinite ground plane is assumed and the stripline feed is modeled as a TEM port. The simulation employs 144 subdomains and the total number of unknowns is approximately 3.3 million. The total memory requirement is 1.2 GB and the factorization time and the solving time per time step are 31 s and 20 s, respectively. The calculated E- and H-plane radiation patterns at 3.75 GHz for various scan angles θ_s and ϕ_s are shown in Fig. 12. It is noted that in all cases the main lobes are steered to the desired angles in both principal planes. The VSWR parameter calculated at the central port is shown in Fig. 13. A prominent feature of the VSWR curve is the two sharp peaks occurring at 3.92 GHz and 5.32 GHz when the array is configured for broadside radiation. Such phenomena, referred to as the impedance anomalies, have been observed in the frequency analysis of infinite array of Vivaldi antennas [30,31]. The anomalies can be modeled as the excitation of certain resonant modes in the unit cell cavities, which are formed by electric conducting surfaces on the side walls and a mag-

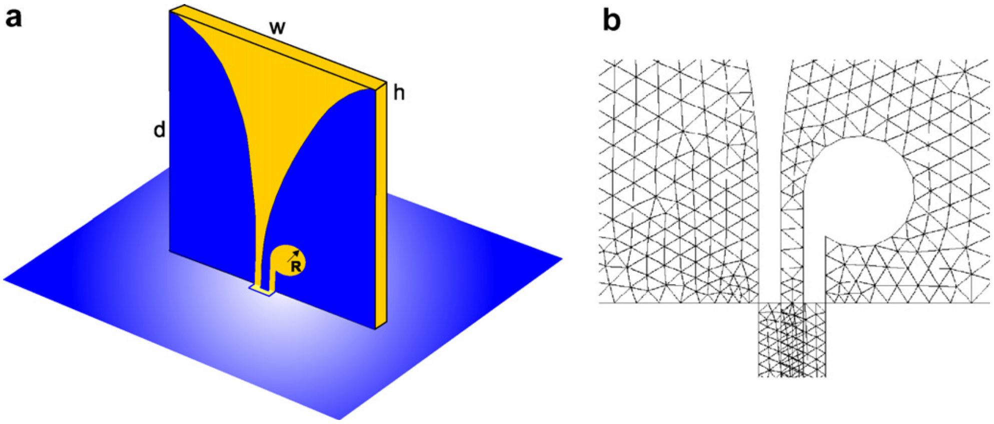

Fig. 10. (a) Vivaldi antenna fed by a stripline. Dimensions: $w = 40$ mm, $d = 55$ mm, $R = 5$ mm. Substrate thickness $h = 1.5$ mm, permittivity $\epsilon_{\text{sub}} = 3.0$. Stripline width $w_1 = 2$ mm, permittivity $\epsilon_{\text{strip}} = 1.0$. (b) Surface mesh in the feed region.

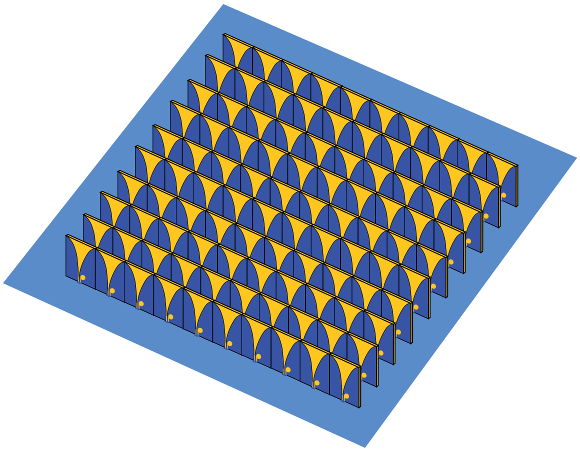

Fig. 11. A 10 × 10 Vivaldi antenna array.

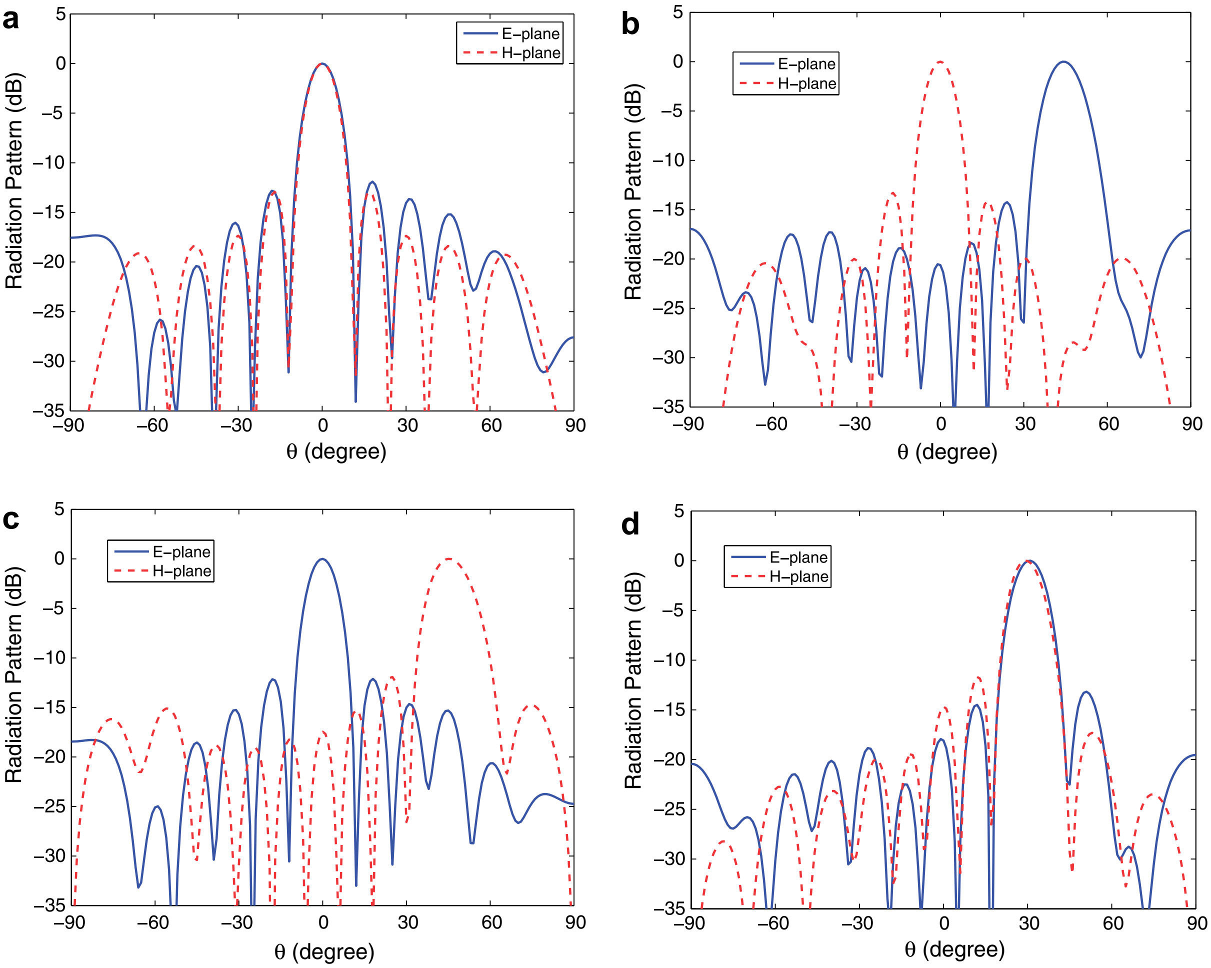

Fig. 12. E- and H-plane radiation pattern of the 10 × 10 Vivaldi array. (a) $\theta_s = 0°$, $\phi_s = 0°$. (b) $\theta_s = 45°$, $\phi_s = 0°$. (c) $\theta_s = 45°$, $\phi_s = 90°$. (d) $\theta_s = 45°$, $\phi_s = 45°$.

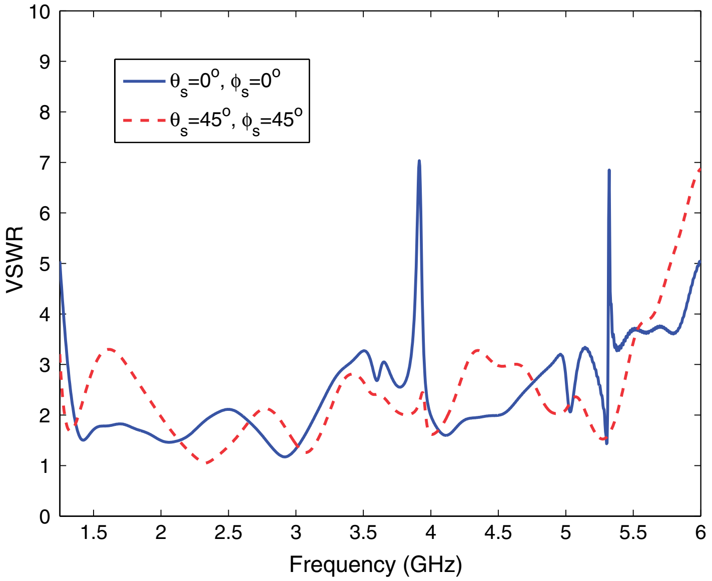

Fig. 13. The VSWR of the 10 × 10 vivaldi array calculated at the central port.

netic conducting surface on the open aperture. The resonant frequencies of the cavities can be approximated by [31]

$$f_{r_{m,n}} = \frac{c_o}{2}\sqrt{\left(\frac{m}{T_x}\right)^2 + \left(\frac{2n+1}{2d}\right)^2} \tag{53}$$

where T_x is the array spacing in the x-direction and d is the height of the substrate. According to (53), the two lowest resonant frequencies of the array are $f_{10} = 3.99$ GHz and $f_{11} = 5.55$ GHz. They are found to be close to the resonant frequencies obtained by the numerical simulation. The cavity model we employed here does not take into account the thin dielectric substrate. If the effect of the dielectrics is considered, the resonant frequencies would become slightly lower. Note that the cavity model (53) is valid only when the array elements are excited identically; hence no obvious resonance pattern is observed when the array is steered to oblique scan angles.

5.3. Dipole in photonic bandgap structures

PBG structures are periodic structures where electromagnetic propagation of certain bands of frequencies is prohibited. This interesting property has inspired many optical applications such as the formation of resonant cavities and guiding structures for optical light. The same EM properties can be observed at microwave frequencies by proper frequency scaling. In microwave regions, the PBG structures are used to suppress surface waves and enhance antenna performance. Here we analyze the radiation of a small dipole antenna inside a PBG structure. The PBG structure is formed by a two-dimensional square array of dielectric rods. We assume the lattice constant of the array is a and choose the radius and length of the rods to by $0.18a$ and $3a$, respectively. The dielectric constant of the rods is set to be 11.56. The dipole is $1.5a$ in length and placed in parallel with the dielectric rods. The TM bandgap of the lattice extends from $f = 0.302c/a$ to $f = 0.443c/a$ [32]. Here we define the TM mode as the mode for which the electric field is parallel to the axis of the dielectric rod. Since the entire structure is symmetric and the plane of symmetry (see Fig. 14b) is effectively a PEC plane for the TM modes, we need to simulate only the upper-half of the computation domain and place a PEC ground plane on the plane of symmetry.

To observe the bandgap properties of the structure, we first simulate the radiation of a dipole inside a PBG cavity. The cavity, shown in Fig. 14a, is formed by removing the central rod in a 9 × 9 array. A dipole antenna

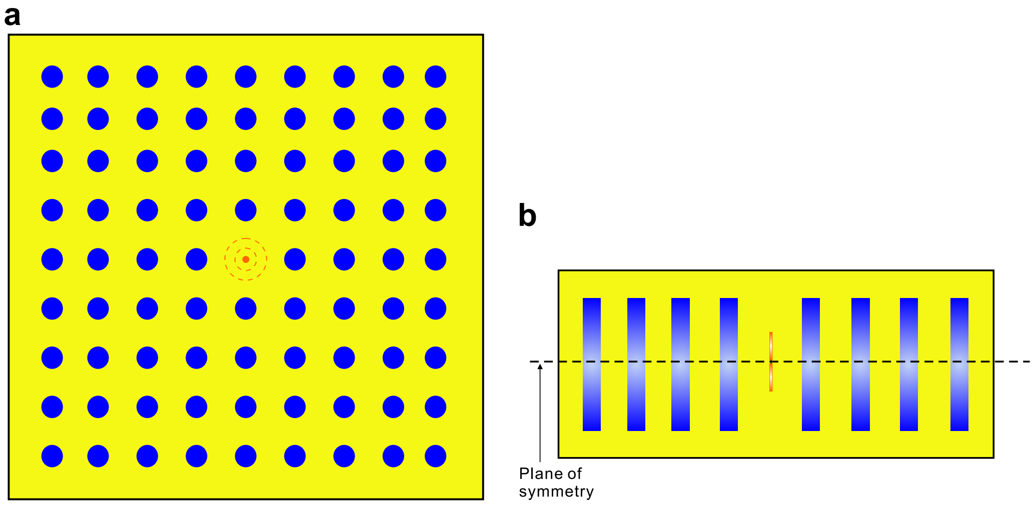

Fig. 14. A 9 × 9 array of dielectric rods with the central rod replaces by a dipole antenna. (a) Top view. (b) Side view.

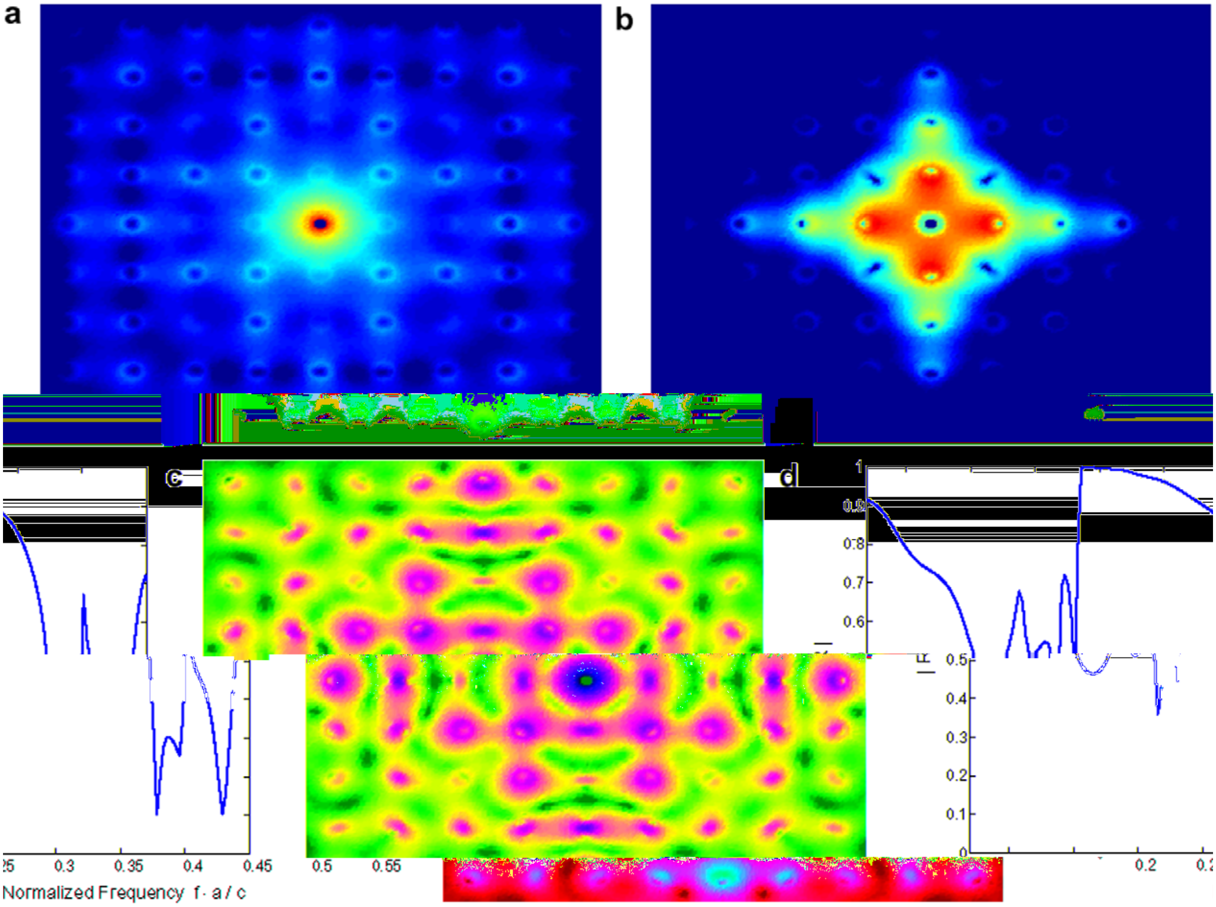

Fig. 15. Magnetic field distribution in the plane of symmetry for various frequencies: (a) $f_1 = 0.25c/a$; (b) $f_2 = 0.35c/a$; (c) $f_3 = 0.5c/a$. (d) Magnitude of the reflection coefficient at the monopole port as a function of frequency.

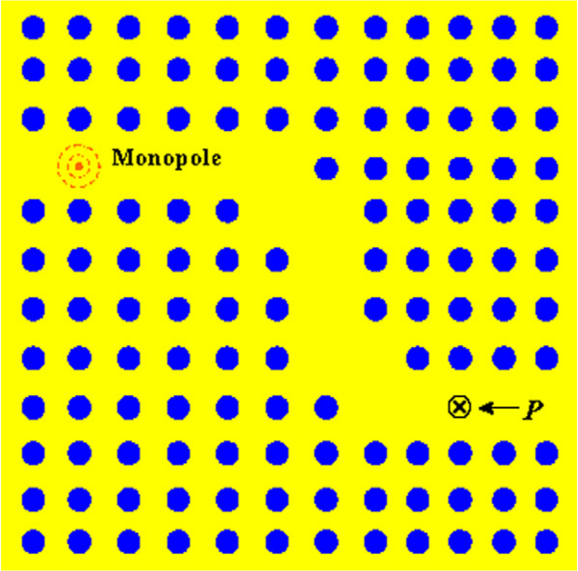

Fig. 16. A PBG waveguide formed by a 12 × 12 array of dielectric rods.

is then placed at the center of the array (which becomes a monopole when the plane of symmetry is modelled as a PEC plane). The antenna launches a modulated Gaussian pulse with a central frequency of $f_0 = 0.4c/a$ into the PBG. We then calculate the magnetic field distribution in the plane of symmetry, shown in Fig. 15a–c, for three frequencies $f_1 = 0.25c/a$, $f_2 = 0.35c/a$, and $f_3 = 0.5c/a$. Since f_2 falls inside the range of

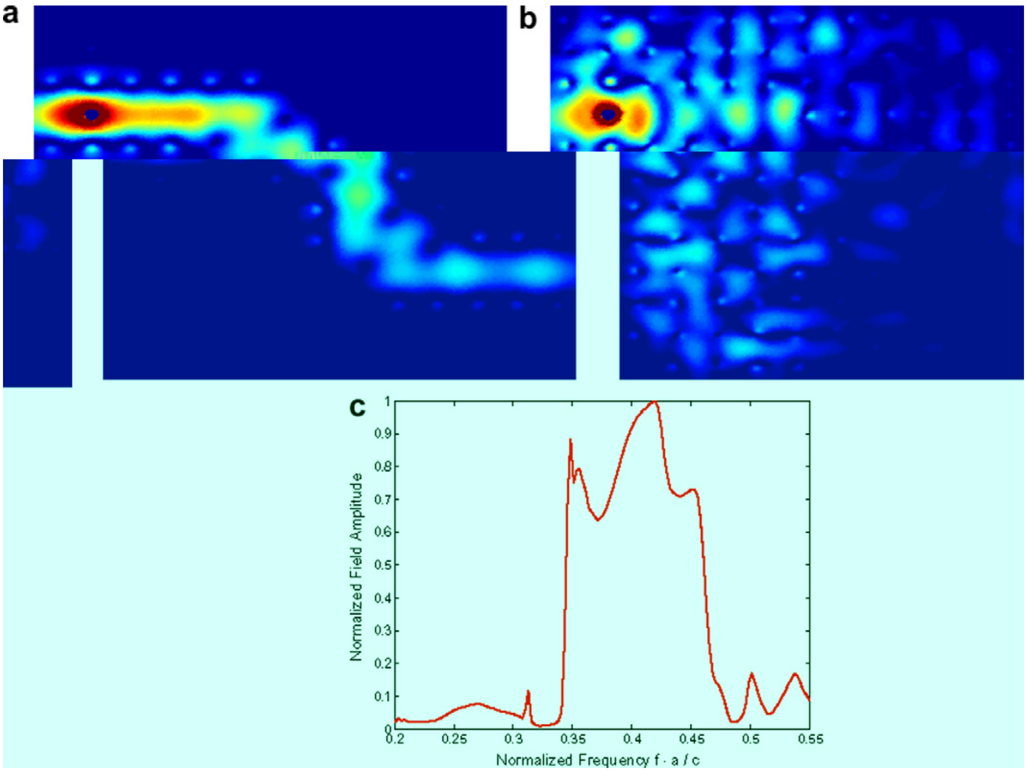

Fig. 17. E_z-field distribution in the plane of symmetry at (a) $f_1 = 0.35c/a$; (b) $f_2 = 0.5c/a$. (c) Normalized field amplitude at a sample point in the waveguide.

the bandgap, the field is confined to the vicinity of the cavity and TM_{11} cavity mode distribution can be clearly seen from Fig. 15b. While in the case of f_1 and f_3, which lie outside the bandgap, the electromagnetic fields are allowed to propagate outside the cavity and thus the field distributions extend the entire computation domain, as seen in Fig. 15a and c. The bandgap can be seen more clearly in Fig. 15d where the reflection coefficient at the monopole port is plotted as a function of frequency.

As a second example, we simulate the radiation of the dipole inside a PBG waveguide. A TM mode waveguide can be created by removing rows or columns of elements from the dielectric rod array. Here we consider a 12×12 array with a total of 17 rods removed to create a double-bended waveguide (shown in Fig. 16). The dipole is placed at the upper-left end of the waveguide and again launches a broadband pulse into the waveguide. We plot the E_z field pattern across the plane of symmetry in Fig. 17a for $f_1 = 0.35c/a$ and Fig. 17b for $f_2 = 0.5c/a$. In Fig. 17a, where the corresponding frequency falls inside the bangap, the propagation of the TM_1 mode along the PBG waveguide is clearly observed, although the magnitude of the propagating mode is reduced at the end of the waveguide due to the reflection losses at the two junctures. In Fig. 17b, however, no obvious guiding phenomenon is observed. Since at this frequency the fields are not confined to the waveguide, the field pattern is merely the result of field scattered by the dielectric rods. Fig. 17c shows the field amplitude as a function of frequency at a sample point P in the waveguide (see Fig. 16), which exhibits a sharp bandgap pattern.

6. Conclusion

In this paper, we proposed using the DFDD–TDFEM for the efficient broadband simulation of large finite arrays. The domain decomposition allows breaking the original computation domain into smaller subdomains, each of which contains a single array element. Then the electromagnetic fields are solved locally inside each subdomain and the fields in adjacent subdomains are related by the surface currents on the subdomain interfaces. By doing so, the original task of solving a large FEM system is avoided. The sizes of the subdomain problems are usually small enough to allow the use of a sparse direct solver. Due to the repetition in the array geometry, only a few subdomain matrices need to be factorized. These factorized matrices are then stored in memory and used repeatedly during the time marching. Since the solving of the subdomain problems at each time step involves only back-substitution, the entire time-marching process is highly efficient.

The proposed DFDD–TDFEM algorithm has several distinct features. First, unlike many other domain-decomposition methods, the proposed method does not require solving a large global problem. All the operations involves in the algorithm are local operations. This is especially advantageous for time-domain simulations where a linear system needs to be solved repeatedly at each time step. As a result, the algorithm is highly efficient and scalable, especially for large-scale computations.

Second, the proposed method is a specialized array solver that fully exploits the repetition of geometry in a typical array configuration. By breaking the original array into identical subdomains and explicitly reusing the identical system matrices, the redundancy in the matrix factorization is removed. As a result, the memory requirement and the CPU time are significantly reduced. The reduction is most dramatic when the array size is large. Although this specialized array solver is not as versatile as a general DFDD algorithm, it is highly efficient for large finite array problems.

Finally, the stability condition of the DFDD–TDFEM algorithm is more relaxed compared to the FDTD and the explicit TDFEM. Instead of the smallest element in the computation domain, the stability condition here depends on the discretization local to the subdomain interfaces. This property is especially beneficial for array configurations where small geometrical features, such as antenna feeds, are confined locally inside the unit cell. In such problems, a much dense spatial discretization can be applied to the local regions without significantly sacrificing the stability condition.

Acknowledgements

This work was supported by a grant from the Air Force Office of Scientific Research via the MURI Program under contract number FA9550-04-1-0326. The authors wish to thank Mr. Yujia Li for calculating the FETI results in Figs. 4–6.

References

[1] J.M. Jin, J.L. Volakis, Scattering and radiation analysis of three-dimensional cavity array via a hybrid finite-element method, IEEE Trans. Antennas Propagat. 41 (1993) 1580–1586.

[2] D.T. McGrath, V.P. Pyati, Phased array antenna analysis with the hybrid finite element method, IEEE Trans. Antennas Propagat. 42 (1994) 1625–1630.

[3] R. Petersson, J. Jin, A three-dimensional time-domain finite element formulation for periodic structures, IEEE Trans. Antennas Propagat. 54 (2006) 12–19.

[4] R. Petersson, J.M. Jin, Analysis of periodic structures via a time-domain finite-element formulation with a Floquet ABC, IEEE Trans. Antennas Propagat. 54 (2006) 933–944.

[5] R. Kindt, K. Sertel, E. Topsakal, J. Volakis, Array decomposition method for the accurate analysis of finite arrays, IEEE Trans. Antennas Propagat. 51 (2003) 1364–1372.

[6] T. Eibert, J. Volakis, Fast spectral domain algorithm for hybrid finite element/boundary integral modeling of doubly periodic structures, IEE Proc. – Microw. Antennas Propag. 147 (2000) 329–334.

[7] J. Yeo, R. Mittra, V. Prakash, Efficient analysis of a class of microstrip antennas using the characteristic basis function method (CBFM), Microwave Opt. Tech. Lett. 39 (2003) 456–464.

[8] S. Lee, M. Vouvakis, J. Lee, A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays, J. Comput. Phys. 203 (2005) 1–21.

[9] Y. Li, J.M. Jin, A vector dual-primal finite element tearing and interconnecting method for solving 3-d large-scale electromagnetic problems, IEEE APS Int. Symp. Dig., Albuquerque, NM, 2006, pp. 3977–3980.

[10] J. Lee, R. Lee, A. Cangellaris, Time-domain finite-element methods, IEEE Trans. Antennas Propagat. 45 (1997) 430–442.

[11] S. Gedney, U. Navsariwala, An unconditionally stable finite element time-domain solution of the vector wave equation, IEEE Microwave Guided Wave Lett. 5 (1995) 332–334.

[12] F. Edelvik, G. Ledfelt, P. Lotstedt, D. Riley, An unconditionally stable subcell model for arbitrarily oriented thin wires in the FETD method, IEEE Trans. Antennas Propagat. 51 (8) (2003) 1797–1805.

[13] D. White, M. Stowell, Full-wave simulation of electromagnetic coupling effects in RF and mixed-signal ICs using a time-domain finite-element method, IEEE Trans. Microwave Theory Tech. 52 (2004) 1404–1413.

[14] D. Jiao, A. Ergin, B. Shanker, E. Michielssen, J. Jin, A fast time-domain higher-order finite-element-boundary-integral method for 3D electromagnetic scattering analysis, IEEE Trans. Antennas Propagat. 50 (9) (2002) 1192–1202.

[15] D. Jiao, J.M. Jin, E. Michielssen, D. Riley, Time-domain finite-element simulation of three-dimensional scattering and radiation problems using perfectly matched layers, IEEE Trans. Antennas Propagat. 51 (2003) 296–305.

[16] D. Riley, J. Jin, Modeling of magnetic loss in the finite-element time-domain method, Microwave Opt. Tech. Lett. 46 (2) (2005) 165–168, July.

[17] Z. Lou, J. Jin, An accurate waveguide port boundary condition for the time-domain finite element method, IEEE Trans. Microwave Theory Tech. 53 (2005) 3014–3023.

[18] Z. Lou, J. Jin, Modeling and simulation of broadband antennas using the time-domain finite element method, IEEE Trans. Antennas Propagat. 53 (2005) 4099–4110.

[19] N. Newmark, A method of computation for structural dynamics, J. Eng. Mechanics Division 85 (1959) 67–94.

[20] C. Farhat, A method of finite element tearing and interconnecting and its parallel solution algorithm, Int. J. Numer. Method. Eng. 32 (6) (1991) 1205–1227, Oct..

[21] C. Farhat, N. Maman, G. Brown, Mesh partitioning for implicit computations via iterative domain decomposition: impact and optimization of the subdomain aspect ratio, Int. J. Numer. Method. Eng. 38 (6) (1995) 989–1000, Mar..

[22] C. Wolfe, U. Navsariwala, S. 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 (2) (2000) 278–284.

[23] M. Vouvakis, and J. Lee, A fast non-conforming DP-FETI domain decomposition method for the solution of large EM problems, IEEE Antennas and Propagation Society International Symposium, vol. 1, pp. 623–626, Monterey, CA, 2004.

[24] U. Navsariwala, S. Gedney, An efficient implementation of the finite-element time-domain algorithm on parallel computers using a finite-element tearing and interconnecting algorithm, Microwave Opt. Tech. Lett. 16 (4) (1997) 204–208, Nov..

[25] L. Margetts, Parallel finite element analysis, Ph.D. thesis, School of Civil Engineering, University of Manchester, 2002.

[26] Z. Lou, J. Jin, A novel dual-field time-domain finite-element domain-decomposition method for computational electromagnetics, IEEE Trans. Antennas Propagat. 54 (2006) 1850–1862.

[27] Z. Lou, Time-domain finite-element simulation of large antennas and antenna arrays, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 2006.

[28] J. Webb, Hierarchal vector basis functions of arbitrary order for triangular and tetrahedral finite elements, IEEE Trans. Antennas Propagat. 47 (1999) 1244–1253.

[29] T. Ho, S. Hart, A broadband coplanar waveguide to slotline transition, IEEE Microwave Guided Wave Lett. 2 (1992) 415–416.

[30] H. Holter, T. Chio, D. Schaubert, Elimination of impedance anomalies in single- and dual-polarized endfire tapered slot phased arrays, IEEE Trans. Antennas Propagat. 48 (2000) 122–124.

[31] Z. Lou, J. Jin, Finite-element analysis of phased-array antennas, Microwave Opt. Tech. Lett. 40 (2004) 490–496.

[32] A. Mekis, J. Chen, I. Kurland, S. Fan, P. Villeneuve, J. Joannopoulos, High transmission through sharp bends in photonic crystal waveguides, IEEE Microwave Guided Wave Lett. 2 (1992) 415–416.