Symmetric Positive Semidefinite FDTD Subgridding Algorithms for Arbitrary Grid Ratios Without Compromising Accuracy

Jin Yan, *Member, IEEE*, and Dan Jiao, *Fellow, IEEE*

***Abstract*— Instability has been a major problem in finite-difference time-domain (FDTD) subgridding methods. Reciprocity has been proposed to overcome the problem but with limited success in producing a symmetric positive semidefinite (SPD) system without compromising accuracy. In this paper, we algebraically derive both 2- and 3-D FDTD subgridding operators, which are SPD by construction, and independent of the grid ratio. Such operators have only nonnegative real eigenvalues, and hence the stability of the resulting explicit time marching is guaranteed. The 3-D operator, the algorithm of which is also applicable to 2-D analysis, further permits the use of a local time step, thus achieving a natural subgridding in both space and time. In addition, the interpretation of the proposed operators in the original FDTD formulation is also provided. Interestingly, not only interface unknowns but also subgrid unknowns are generated in a different way, as compared to the original FDTD, to simultaneously obtain an SPD system and to ensure accuracy. Extensive numerical simulations have demonstrated the accuracy, stability, and efficiency of the proposed new subgridding algorithms.**

***Index Terms*— Finite-difference time-domain method (FDTD), positive semidefinite, spatial subgridding, stability, subgridding, symmetric, temporal subgridding.**

I. Introduction

SUBGRIDDING is an effective means to locally refine the grid in a finite-difference time-domain (FDTD) simulation. This is especially useful when simulating a multiscale problem. In literature, many FDTD subgridding methods have been developed, such as the variable step size method [1], the mesh refinement algorithm [2], the multigrid methods [3], and subgridding schemes with separated temporal and spatial interfaces [4]. It has been observed that stability and accuracy are two major challenges, which are also two competing factors. One can always develop very accurate interpolation techniques to obtain the field unknowns at the interface between a subgrid and a base grid. However, the stability of the resultant time marching cannot be guaranteed. This is because the system matrices resulting from these interpolation schemes are often unsymmetrical. An unsymmetrical matrix can support complex-valued and even negative eigenvalues, which will render an explicit FDTD time marching absolutely unstable. A proof on this point can be found in [5]. Various approaches have been proposed to remedy this issue [3], [6]–[8]. To ensure stability, enforcing reciprocity of the fields has been proposed [9]–[11]. However, it is difficult to preserve the symmetry of the original FDTD for general subgrid arrangements and grid ratios, without compromising accuracy. In an intelligently designed method like [4], the subgrid must also be arranged in a special way, certain grid ratios should be chosen, the **H** is not centered in the **E** loop, the **E** may not be sampled from the edge center, etc., in order to ensure the resultant numerical system to be symmetric.

In this paper, we propose a systematic approach to make FDTD subgridding symmetric positive semidefinite (SPD) regardless of the grid ratio and arrangement. We first rewrite the FDTD into a single-grid patch-based formulation. This formulation facilitates us to find out how the matrix equations in different domains are assembled to form a global matrix equation in the FDTD. It turns out to be very different from other methods. Based on this finding, we algebraically derive an FDTD subgridding operator that is theoretically SPD. We also translate the operator from its matrix form back to the original form of the FDTD to help readers better understand how the subgrid unknowns are handled in the proposed method. Interestingly, not only the subgrid interface unknowns are changed in their way of being generated, but also the unknowns inside the subgrid. The preliminary work of this paper is published in our conference paper [12] as a 2-D subgridding operator. In this paper, in addition to providing a systematic algebraic approach to derive an SPD operator, and completing the development of the 2-D subgridding algorithm from all theoretical aspects, we further develop an SPD 3-D operator. This 3-D operator ensures accuracy by preserving the original FDTD's differencing schemes, where normal **H** is obtained from the center of an **E**'s loop, and vice versa. In addition, a local time stepping is also permitted by this 3-D subgridding operator. Besides theoretically guaranteed stability and accuracy (which preserves the second-order accuracy of the original FDTD in time and space), the implementation of the proposed methods is also made straightforward. It should be noted that the proposed subgridding does not compromise

Manuscript received July 1, 2017; revised September 20, 2017; accepted October 15, 2017. Date of publication November 27, 2017; date of current version December 12, 2017. This work was supported in part by the NSF under Grant 1619062 and in part by DARPA under Grant HR0011-14-1-0057. This paper is an expanded version from the 2017 IEEE MTT-S International Microwave Sympoisum Conference, Honolulu, HI, USA, June 4–9, 2017. *(Corresponding author: Dan Jiao.)*

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: djiao@purdue.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMTT.2017.2766071

accuracy in the sense that every unknown in the grid and subgrid is solved using the same second-order accuracy as that in the conventional FDTD.

This paper is organized as follows. In Section II, we give a brief review of the background of this paper, including a patch-based single-grid FDTD representation and the stability analysis of an FDTD subgridding scheme. In Section III, we present a systematic algebraic approach to derive an SPD subgridding operator. In Section IV, we develop a 2-D SPD subgridding operator based on the approach of Section III. In Section V, a 3-D SPD operator is developed, which also naturally permits local time stepping. In Section VI, numerical examples are given to validate the proposed new subgridding algorithms. Section VII relates to our conclusions.

II. Preliminaries

A. Review of the Patch-Based Single-Grid Representation of the FDTD

To facilitate the development of a generic symmetric subgridding algorithm, we employ a patch-based single-grid formulation of the FDTD recently developed in [13] and [14]. This formulation helps reveal more clearly how the equations in one domain are stitched with the equations in the other domain to form a global system in the FDTD algorithm.

Consider a 2- or 3-D grid. Let the number of **E** and **H** unknowns be, respectively, N_e, and N_h. Let $\{e\}$ be a global electric field unknown vector, while $\{h\}$ being a global magnetic field unknown vector. The original FDTD can be written into the following form:

$$\mathbf{S}_e\{e\} = -\mathbf{D}_\mu\{\dot{h}\} \tag{1}$$

$$\mathbf{S}_h\{h\} = \mathbf{D}_\epsilon\{\dot{e}\} + \mathbf{D}_\sigma\{e\} + \{j\} \tag{2}$$

where a single dot above a letter denotes a first-order time derivative, $\{j\}$ represents a current source vector, and $\mathbf{D}_\mu$, $\mathbf{D}_\sigma$, and $\mathbf{D}_\epsilon$ are diagonal matrices of permeability, conductivity, and permittivity, respectively.

Based on the patch-based single-grid representation in [14], each row of $\mathbf{S}_e$ in (1) corresponds to one patch in the grid, and hence one magnetic field. This field is located at the patch center and normal to the patch. Each row of $\mathbf{S}_e$ has at most four nonzero entries, at the columns corresponding to the electric fields along the four sides of the patch. Take an arbitrary ith row, $\mathbf{S}_e^{(i)}$, as an example, it can be written as

$$\mathbf{S}_e^{(i)} = \left[-{L_i}^{-1}, {L_i}^{-1}, {W_i}^{-1}, -{W_i}^{-1} \right] \oplus \text{zeros}(1, N_e) \tag{3}$$

where $\oplus$ denotes an extended addition based on the global indexes of the four local **E** unknowns of patch i, and L_i and W_i are, respectively, the two side lengths of patch i. Obviously, $\mathbf{S}_e^{(i)}\{e\}$ is nothing but a curl of **E** operation in patch i, which produces $-\mu\dot{h}_i$, and h_i is the normal magnetic field at the patch center.

In a uniform grid, $\mathbf{S}_h$ in (2) is simply the transpose of $\mathbf{S}_e$, that is,

$$\mathbf{S}_h = {\mathbf{S}_e}^T. \tag{4}$$

In other words, the ith column of $\mathbf{S}_h$ is the same as the ith row of $\mathbf{S}_e$

$$\mathbf{S}_h^{(i)} = \left[-{L_i}^{-1}, {L_i}^{-1}, {W_i}^{-1}, -{W_i}^{-1} \right]^T \oplus zeros(N_e, 1). \tag{5}$$

Hence, in $\mathbf{S}_h$, each column corresponds to one patch, and thereby one magnetic field. $\mathbf{S}_h$ is thus of dimension N_e by N_h. A column i of $\mathbf{S}_h$ has also at most four nonzero entries, located at the rows corresponding to the four electric fields of patch i. The column signifies how the h_i on the patch is used to produce the electric fields along the four sides of the patch.

Eliminating $\{h\}$ from (1) and (2), we obtain

$$\mathbf{D}_\epsilon\{\ddot{e}\} + \mathbf{D}_\sigma\{\dot{e}\} + \mathbf{C}\{e\} = -\{\dot{j}\} \tag{6}$$

where two dots above a letter denote a second-order time derivative, and $\mathbf{C}$ can be written as

$$\mathbf{C} = \mathbf{S}_h\mathbf{D}_{\mu^{-1}}\mathbf{S}_e = \sum_{i=1}^{N_h} {\mu_i}^{-1}\left(\mathbf{S}_h^{(i)}\right)_{N_e\times 1}\left(\mathbf{S}_e^{(i)}\right)_{1\times N_e} \tag{7}$$

which is the summation of the rank-1 matrix over all of the N_h patches. This rank-1 matrix is obtained from multiplying the column vector $\mathbf{S}_h^{(i)}$ on patch i by the row vector $\mathbf{S}_e^{(i)}$.

Using the single-grid patch-based formulation [14], to perform an FDTD simulation, we only need to loop over all the patches in a single grid, regardless of whether the grid is 2- or 3-D. For each patch, we generate one row vector $\mathbf{S}_e^{(i)}$ shown in (3), and one column vector $\mathbf{S}_h^{(i)}$ shown in (5). Multiplying the two together and add the resultant rank-1 matrix of each patch, we obtain global $\mathbf{C}$. The time-domain simulation can then be performed based on (6).

B. Stability of FDTD Subgridding Algorithms

The stability of the FDTD is governed by $\mathbf{C}$. If $\mathbf{C}$ is unsymmetrical, in general it supports complex or negative eigenvalues. When these eigenvalues exist, an explicit time marching is absolutely unstable, which has been proven in [5] (Section IV of this reference paper). This is why many FDTD subgridding algorithms cannot ensure stability. For convenience of readers, we provide a brief summary of the proof in [5] here.

The FDTD leapfrog-based time marching is equivalent to a central-difference-based time marching of (6). Performing a z-transform of the central-difference-based marching of (6) results in the following equation:

$$(z-1)^2 + \Delta t^2\lambda z = 0 \tag{8}$$

where λ is the eigenvalue of $\mathbf{D}_\epsilon^{-1}\mathbf{C}$, and Δt is the time step. The two roots of (8) can be readily found as

$$z_{1,2} = \frac{2 - \Delta t^2\lambda \pm \sqrt{\Delta t^2\lambda(\Delta t^2\lambda - 4)}}{2}. \tag{9}$$

If $\mathbf{C}$ is not symmetrical, its eigenvalues either are real (can be negative) or come in complex-conjugate pairs. For complex-valued λ as well as negative ones, the two roots z_1 and z_2 shown in (9) satisfy $z_1 z_2 = 1$, and neither of them has modulus equal to 1. As a result, the modulus of one of them must be greater than 1, and hence an explicit time marching of (6) must

be unstable. If $\mathbf{C}$ is SPD, its eigenvalues are nonnegative real. In this case, the stability of an explicit marching is guaranteed if we choose

$$\Delta t \leq 2/\sqrt{\lambda_{\max}} \tag{10}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{D}_\epsilon^{-1}\mathbf{C}$ [15]. Obviously, the symmetric positive semidefiniteness of $\mathbf{C}$ is satisfied in the original FDTD in a uniform grid, because of (4).

III. Proposed Algebraic Approach for Developing an SPD Subgridding Algorithm

In this section, we first analyze how the FDTD equations in different domains are stitched together to perform a global simulation in the entire grid. We then algebraically combine the equations in the subgrids with those in the base grid, and systematically derive a subgridding operator that can be symmetric and positive semidefinite by construction.

A. On the Assembling Mechanism of the FDTD

By using (7), we now can analyze how the submatrices in different domains are actually stitched in the FDTD. Since each row (column) vector of $\mathbf{S}_e^{(i)}$ ($\mathbf{S}_h^{(i)}$) is built based on the global ordering of the $\mathbf{E}$ unknowns, the field tangential continuity at the patch interface is actually used to assemble $\mathbf{C}$ from each patch's contribution. However, the diagonal matrices in front of the second- and first-order time derivatives in (6) are assembled differently. Shall they be assembled in the same way as $\mathbf{C}$ is assembled from each patch, the resultant matrices would be different since each edge is shared by more than one patch. This is very different from other methods such as the finite element method.

To better understand the assembling mechanism, we can consider two regions. Based on (7), the equations in the two regions are combined in the following way:

$$\mathbf{D}_\epsilon\{\ddot{e}\} = -\left[\mathbf{S}_{h,1}\mathbf{D}_{\mu_1^{-1}}\mathbf{S}_{e,1}\{e\} + \mathbf{S}_{h,2}\mathbf{D}_{\mu_2^{-1}}\mathbf{S}_{e,2}\{e\}\right] \tag{11}$$

where $\mathbf{S}_{h,1(2)}$ is composed of all the column vectors generated from the patches in region 1 (2); $\mathbf{S}_{e,1(2)}$ is composed of all the row vectors from the patches in region 1 (2). Notice that each column vector in $\mathbf{S}_{h,1(2)}$ is of full length N_e, and the same is true for the row vectors in $\mathbf{S}_{e,1(2)}$. The addition in (11) is a true addition instead of assembling. Here, source term is omitted as it is irrelevant to stability. $\mathbf{D}_\sigma$ term is also ignored for simplicity. It can be added without any difficulty.

From (11), it can be seen that the solution of an $\mathbf{E}$ unknown is the *addition* of two contributions. One is from region 1, shown by the first term. This term is nothing but $\mathbf{S}_{h,1}h_1$'s time derivative, where h_1 is the magnetic field at the e's patch in region 1. The other is from region 2, shown by the second term in (11). This term is nothing but $\mathbf{S}_{h,2}h_2$'s time derivative, where h_2 is the magnetic field at the e's patch in region 2. If we denote

$$-\mathbf{S}_{h,1}\mathbf{D}_{\mu_1^{-1}}\mathbf{S}_{e,1}\{e\} = \mathbf{D}_\epsilon\{\ddot{e}\}_+ \tag{12}$$

$$-\mathbf{S}_{h,2}\mathbf{D}_{\mu_2^{-1}}\mathbf{S}_{e,2}\{e\} = \mathbf{D}_\epsilon\{\ddot{e}\}_- \tag{13}$$

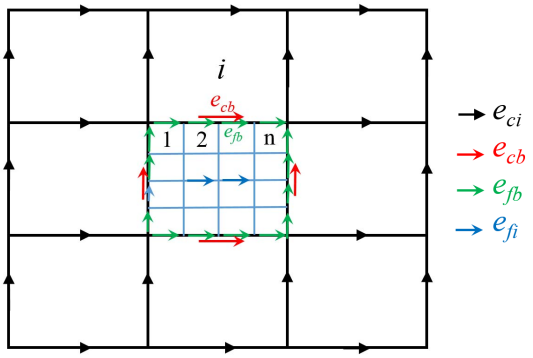


Fig. 1. Subgrid embedded in a 2-D base grid.

as can be seen in (11), the assembling of two-domain equations is performed by

$$\{e\} = \{e\}_+ + \{e\}_-. \tag{14}$$

In other words, neither of the two terms represents a complete e. The two must be added to complete e's solution. If both of them represent a complete e, when adding (12) upon (13), we would get $2\{e\}$ in the right-hand side, which violates (11). The addition shown in (14), in fact, completes a curl $\mathbf{H}$ operation across the patches sharing the $\mathbf{E}$ unknown. One can use a single edge shared by two patches in a 2-D grid to see this point more clearly.

Now if the two regions are a base-grid region, and a subgrid region, respectively, as illustrated in Fig. 1, the same principle applies to add the equations generated from each region. However, certain transformation between unknowns is needed since the mesh in the subgrids is different from that in the base grid.

B. Proposed Algebraic Approach

After finding out how subdomain equations are added up in the FDTD, we proceed to algebraically derive an SPD operator for subgridding. We denote the base-grid region by c, standing for the $c$oarse mesh in the base grid; and the subgrid region by f, representing a finer mesh in the subgrid.

Let

$$e_f = \{e_{\text{fb}}\ e_{\text{fi}}\}^T \tag{15}$$

be a vector containing all the $\mathbf{E}$ unknowns in the subgrid, where e_{fb} are at the boundary between the subgrid and the base grid, and e_{fi} are internal to the subgrid. Take a 2-D subgrid shown in Fig. 1 as an example, the e_{fb} edges are marked in green, while e_{fi} edges are blue. The same subdivision is done in a 3-D grid.

Let

$$e_c = \{e_{\text{ci}}\ e_{\text{cb}}\}^T \tag{16}$$

be an $\mathbf{E}$ unknown vector in the base grid, where e_{cb} are on the subgrid boundary, and e_{ci} are internal to the base grid. They are shown as edges marked with red and black arrows, respectively, in Fig. 1. The e_{ci} are certainly separated from e_{fi}. However, e_{cb} and e_{fb} overlap, as can be seen in Fig. 1.

Since the field tangential continuity at the subgrid interface should be satisfied, one common set of **E** unknowns should be used at the interface. If we use $e_{\rm cb}$ as the common set, then we have a global **E** unknown vector as the following:

$$\{e\}_{N_e\times 1} = \{e_{\rm ci}\ \mathbf{e}_{\rm cb}\ e_{\rm fi}\}^T \tag{17}$$

to combine the equations in two regions. The total number of **E** unknowns, N_e is, hence,

$$N_e = \#e_{\rm ci} + \#e_{\rm cb} + \#e_{\rm fi} \tag{18}$$

that is, the total number of e_c unknowns and subgrid unknowns internal to the subgrid. This is in contrast to using the following $\{e_{\rm sub}\}$ as the global **E** unknown vector:

$$\{e_{\rm sub}\}_{N_{e,s}\times 1} = \{e_{\rm ci}\ \mathbf{e}_{\rm fb}\ e_{\rm fi}\}^T \tag{19}$$

where the $e_{\rm fb}$ unknowns are adopted on the subgrid interface instead of $e_{\rm cb}$. Here, the length of $\{e_{\rm sub}\}$, $N_{e,s}$, is the sum of e_f unknown number and the number of base-grid unknowns internal to the base grid.

Using (17) as the global **E** unknown vector, based on (12), the e's solution contributed from the base grid can be written as

$$\mathbf{D}_\epsilon\{\ddot{e}\}_+ = -\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c}\{e\} \tag{20}$$

where $\mathbf{S}_{e,c}$ is obtained from all patches in the based grid, whose ith row has the expression shown in (3). The $\mathbf{S}_{h,c}$ is simply the transpose of $\mathbf{S}_{e,c}$. As for the contribution from the subgrid, we first build the subgrid system using $\{e_{\rm sub}\}$, i.e., subgrid **E** unknowns. We obtain

$$\mathbf{D}_{\epsilon,f}\{\ddot{e}_{\rm sub}\}_- = -\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\{e_{\rm sub}\} \tag{21}$$

where each row of $\mathbf{S}_{e,f}$ is written for one fine patch in the subgrid, and $\mathbf{S}_{h,f}$ is $\mathbf{S}_{e,f}$'s transpose.

To combine (21) with (20), we need to transform (21) from an $\{e_{\rm sub}\}$-based system to a new one based on $\{e\}$. Since $e_{\rm cb}$ and $e_{\rm fb}$ are both on the subgrid boundary, one can be represented by the other. Let

$$\{e_{\rm fb}\} = \mathbf{P}_{\rm fc}\{e_{\rm cb}\} \tag{22}$$

where $\mathbf{P}_{\rm fc}$ is a mapping matrix used to obtain $e_{\rm fb}$ from $e_{\rm cb}$. Using (22), we can build a global mapping matrix

$$\mathbf{P}_{N_{e,s}\times N_e} = \begin{bmatrix} (\mathbf{I})_{\#e_{\rm ci}\times\#e_{\rm ci}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{P}_{\rm fc})_{\#e_{\rm fb}\times\#e_{\rm cb}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (\mathbf{I})_{\#e_{\rm fi}\times\#e_{\rm fi}} \end{bmatrix} \tag{23}$$

such that

$$\{e_{\rm sub}\}_{N_{e,s}\times 1} = \mathbf{P}\{e\}. \tag{24}$$

As a result, (21) formed for $\{e_{\rm sub}\}$ can be transformed to

$$\mathbf{D}_{\epsilon,f}\mathbf{P}\{\ddot{e}\}_- = -\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\{e\}. \tag{25}$$

However, by doing so, the right-hand side matrix of (25) is not symmetric any more. To obtain symmetry, we can multiply both sides of (25) by $\mathbf{P}^T$, obtaining

$$\mathbf{P}^T\mathbf{D}_{\epsilon,f}\mathbf{P}\{\ddot{e}\}_- = -\mathbf{P}^T\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\{e\}. \tag{26}$$

Though symmetric, the above cannot be added upon (20) yet since $\{e\}_+ \neq \{e\}_-$. To make the addition like (14) feasible, further conversion is required.

Since the n subgrid unknowns $e_{\rm fb}$ and the $e_{\rm cb}$ are on the same base-grid edge, they share the same permittivity in common. Thus,

$$\mathbf{P}^T\mathbf{D}_{\epsilon,f}\mathbf{P} = \mathbf{P}^T\mathbf{P}\mathbf{D}_\epsilon \tag{27}$$

and hence (26) can be rewritten as

$$\mathbf{D}_\epsilon\{\ddot{e}\}_- = -(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\{e\}. \tag{28}$$

Now, based on (14), we can add (28) upon (20) to complete the entire system of equations, which is

$$\mathbf{D}_\epsilon\{\ddot{e}\} = -\left[\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c} + (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\right]\{e\}. \tag{29}$$

Obviously, the matrix contributed from the subgrid region is a symmetric matrix front multiplied by the inverse of a symmetric matrix $\mathbf{P}^T\mathbf{P}$. This subgrid matrix can be further made symmetric. In the next two sections, we show detailed constructions for 2- and 3-D problems, respectively.

IV. Proposed 2-D SPD Subgridding Algorithm

A. 2-D SPD Subgridding Operator

For a 2-D grid, consider an arbitrary grid ratio n, there are n subgrid boundary unknowns $e_{\rm fb}$ along the edge of a single $e_{\rm cb}$, as illustrated in Fig. 1. $\mathbf{P}_{\rm fc}$ can be simply chosen as

$$\mathbf{P}_{\rm fc} = [1\ 1\ 1\ \ldots\ 1]^T_{n\times 1} \tag{30}$$

which is a column vector containing n ones. This means the electric field at the base-grid edge center is used as the electric field along the edge.

To derive an SPD system from (29), first, we realize $\left(\mathbf{P}^T\mathbf{P}\right)^{-1}$ is diagonal, having the following entries:

$$(\mathbf{P}^T\mathbf{P})^{-1} = \begin{bmatrix} \mathrm{diag}\{1\}_{\#e_{\rm ci}} & 0 & 0 \\ 0 & \mathrm{diag}\{d^{-1}\}_{\#e_{\rm cb}} & 0 \\ 0 & 0 & \mathrm{diag}\{1\}_{\#e_{\rm fi}} \end{bmatrix} \tag{31}$$

where # denotes the cardinality of a set, and diag$\{\cdot\}$ denotes a diagonal matrix of element $(\cdot)$

$$\mathbf{P}_{\rm fc}^T\mathbf{P}_{\rm fc} = \mathrm{diag}\{d\} \tag{32}$$

is diagonal, and for the choice of (30)

$$d = n. \tag{33}$$

In (31), if we change the diagonal entries corresponding to $e_{\rm ci}$ from 1 to d^{-1}, there is no change of (29) as the matrix of the subgrid does not involve $e_{\rm ci}$. If we also change the diagonal entries corresponding to $e_{\rm fi}$ from 1 to d^{-1}, we only need to scale the left-hand side of (29) corresponding to $e_{\rm fi}$ in the same way. This is because the base-grid matrix does not involve $e_{\rm fi}$ unknowns. As a result, (29) can be rewritten as

$$\mathbf{D}_p\mathbf{D}_\epsilon\{\ddot{e}\} = -\mathbf{C}_{\rm new}\{e\} \tag{34}$$

in which

$$\mathbf{D}_p = \mathrm{diag}\{\{1\}_{\#e_{\rm ci}}, \{1\}_{\#e_{\rm cb}}, \{d^{-1}\}_{\#e_{\rm fi}}\} \tag{35}$$

and

$$\mathbf{C}_{\text{new}} = \mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c} + d^{-1}\mathbf{P}^T\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}. \quad (36)$$

Obviously, the above numerical system formulated for subgridding is governed by a symmetric positive definite eigenvalue problem of

$$\mathbf{D}_p\mathbf{D}_\epsilon v = \lambda \mathbf{C}_{\text{new}} v. \quad (37)$$

Since $\mathbf{C}_{\text{new}}$ is SPD, and diagonal matrix $\mathbf{D}_p\mathbf{D}_\epsilon$ is positive definite, the eigenvalues of (37) must be nonnegative real. Hence, the stability of the proposed subgridding algorithm is guaranteed.

Equation (36) is written for one subgrid region, if there are multiple subgrid regions, (36) becomes

$$\mathbf{C}_{\text{new}} = \mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c} + \sum_{j=1}^{p} d_j^{-1}\mathbf{P}_j^T\mathbf{S}_{h,j}\mathbf{D}_{\mu_j^{-1}}\mathbf{S}_{e,j}\mathbf{P}_j \quad (38)$$

where p is the number of subgrid regions, and j denotes the subgrid index. $\mathbf{C}_{\text{new}}$ clearly remains to be SPD. The left-hand side diagonal matrix $\mathbf{D}_p$ in (34) will also be augmented with d_i for each subgrid region accordingly.

B. Regarding the Computational Overhead

The implementation of (36) and (38) is, in fact, straightforward in the patch-based FDTD formulation. Basically, for every patch in the subgrid, after we obtain its row vector $\mathbf{S}_e^{(i)}$ of four nonzero entries, we simply extend it to length N_e based on the global ordering of local $\mathbf{E}$ unknowns. The operation of $\mathbf{S}_e^{(i)}\mathbf{P}$ based on the mapping matrix shown in (30) is simply to place every entry of $\mathbf{S}_e^{(i)}$ whose column is an e_{fb} unknown at the column of corresponding e_{cb} unknown. The $\mathbf{P}^T\mathbf{S}_h$ is nothing but the transpose of $\mathbf{S}_e\mathbf{P}$. The d_j entry can also be readily evaluated based on (30). Hence, the computational overhead of the proposed subgridding scheme is negligible. In other words, the computation additional to the conventional FDTD is negligible.

C. Use of Average Length for e_{cb} and e_{fb} Unknowns

For an electric field unknown e at the boundary between the subgrid and the base grid, the magnetic fields at the two patches sharing the e edge are used to obtain e. These e include e_{cb} and e_{fb}. The two patches sharing such a boundary e have different sizes, as one is in the subgrid, and the other in the base grid.

Like what is done in a nonuniform FDTD grid, for better accuracy, we use an average length L_{ave} of the two patches along the direction perpendicular to the e edge to perform the curl $\mathbf{H}$ operation to obtain e. To do so, the row entry of $\mathbf{S}_h^{(i)}$ of the two patches corresponding to the e field will be changed from $1/L_i$ shown in (3) to $1/L_{\text{ave}}$. The $\mathbf{S}_e^{(i)}$ would not be modified since the $\mathbf{H}$ point is still centered by the four $\mathbf{E}$ edges. In this way, although the resulting $\mathbf{S}_h^{(i)}\mathbf{S}_e^{(i)}$ is not symmetric for the patch involved in the modification, the end result can still be made symmetric as shown in the following.

After replacing L_i by L_{ave} for all the boundary e fields, the change of $\mathbf{C}_{\text{new}}$ is simply

$$\mathbf{C}_{\text{new}} = \mathbf{D}_1\left(\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c} + d^{-1}\mathbf{D}_2\mathbf{P}^T\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\right) \quad (39)$$

where the two diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ are, respectively,

$$\mathbf{D}_1 = \text{diag}\{\{1\}_{\#e_{\text{ci}}}, \{L_c/L_{\text{ave}}\}_{\#e_{\text{cb}}}, \{1\}_{\#e_{\text{fi}}}\} \quad (40)$$

and

$$\mathbf{D}_2 = \text{diag}\{\{1\}_{\#e_{\text{ci}}}, \{L_f/L_c\}_{\#e_{\text{cb}}}, \{1\}_{\#e_{\text{fi}}}\} \quad (41)$$

where L_c is the base-grid cell size, and L_f is subgrid cell size. The matrix in the parenthesis of (39) appears to be unsymmetrical. However, again, we can change $\mathbf{D}_2$ to a diagonal matrix of L_f/L_c, i.e., $1/n$. Hence, (39) becomes

$$\mathbf{C}_{\text{new}} = \mathbf{D}_1\left(\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c} + d^{-1}n^{-1}\mathbf{P}^T\mathbf{S}_{h,f}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\right). \quad (42)$$

Meanwhile, we scale the left-hand side $\mathbf{D}_p$ of (34) by the same $1/n$ for entries corresponding to e_{fi} so that e_{fi}'s solutions are not changed. As a result, the final numerical system remains to have nonnegative real eigenvalues.

D. Transforming Back to the Original FDTD Formulation and Accuracy Verification

The operator derived in the above can also be translated back to the original FDTD format to understand its meaning, and verify its accuracy. The key of a subgridding method is essentially how the interface unknown e_{cb} is generated and its impact on symmetry, since the generation of other unknowns can stay unchanged.

Consider an arbitrary e_{cb} unknown shown in Fig. 1, whose global index is k. It is shared by patch i in the base grid, and n fine patches in the subgrid denoted from 1 to n. From (34) and (36), we obtain

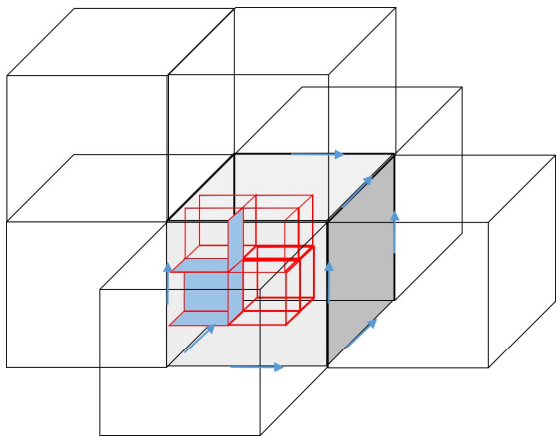
$$\epsilon\{e_{\text{cb}}^{\cdot}\} = \left(\mathbf{S}_h^{(i)}\right)_k h_i + 1/n\sum_{j=1}^{n}\left(\mathbf{S}_h^{(j)}\right)_k h_j \quad (43)$$

where subscript k denotes the kth row entry of the vector in the parenthesis. The above is nothing but to do

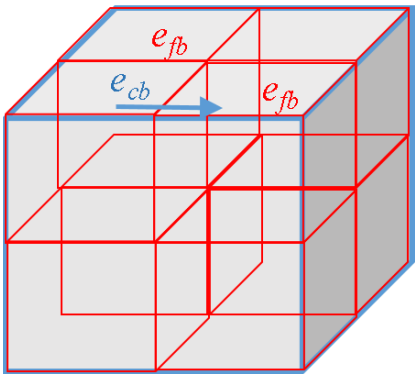
$$\epsilon\{e_{\text{cb}}^{\cdot}\} = \frac{h_i}{L_k} - \frac{(h_1 + h_2 + \cdots + h_n)}{nL_k} \quad (44)$$

which is to use the average of the magnetic fields generated from the n subgrid patches to obtain e_{cb}, and hence accurate. In (44), L_k is the length parameter associated with edge k. After the treatment described in Section IV-C, L_k is averaged from the side lengths of the two patches sharing edge k. Our numerical experiments also show this treatment is important to obtain good accuracy.

As for the impact of (44) on symmetry, as can be seen, the coefficient in front of the base grid h_i is the same as its normal value. However, the coefficient of the h_j for the n fine patches is scaled by $1/n$. This will make the coupling coefficient from h_j to e_{cb} and thereby e_{fb} different from that of e_{fb} to h_j, yielding an unsymmetrical matrix. However, as shown by (34), the proposed method also multiplies all the rows corresponding to e_{fi} by $1/n$. This will make the



(a)



(b)

Fig. 2. (a) Subgrid embedded in a 3-D base grid and its patch assignment. (b) Subgrid region.

coupling coefficient from h_j to all the four e in patch j scaled simultaneously by $1/n$, and hence the resulting system remains symmetric. In other words, the original $\mathbf{S}_h^{(i)}\mathbf{S}_e^{(i)}$ is symmetric. If the rows of $\mathbf{S}_h^{(i)}$ are selectively scaled, the resultant would not be symmetric. However, if all rows are scaled by the same coefficient, the product will remain symmetric, as it becomes simply $1/n\mathbf{S}_h^{(i)}\mathbf{S}_e^{(i)}$. Although in this way, the e_{fi} solved would become different, but as shown in the proposed method, it can be changed back by scaling the left-hand side matrix in the same way. As can be seen, the aforementioned symmetric operations are difficult to be conceived in the local differencing-based framework of the original FDTD, but it becomes feasible using the proposed system-level approach.

V. Proposed 3-D SPD Subgridding Algorithm

For 3-D problems, the formulation derived in Section III remains the same. This is because a 3-D grid can be viewed as an assembly of multiple rectangular patches. Each patch's normal direction can be orientated in any of the x-, y-, and z-directions. The $\mathbf{S}_e^{(i)}$ row vector and $\mathbf{S}_h^{(i)}$ column vector for an arbitrary patch i have the same expressions as those in 2-D. However, the e_{fb} would not only reside along the coarse edges at the subgrid interface, i.e., overlapping with e_{cb}, but also appear on the surfaces of the subgrid interface. This set of surface e_{fb} unknowns are not shared in common by base grid and subgrid. The $\mathbf{P}$, thus, has to be carefully designed to ensure the same SPD property without affecting accuracy. In this section, we first propose a subgridding algorithm for 3-D analysis, which also permits the use of a local time step only dependent on the local grid size regardless of the grid size elsewhere. We then show how the new 3-D algorithm is related to the subgridding operator developed in Section III.

A. Proposed 3-D Subgridding Algorithm

Now, consider a subgrid present in a 3-D base grid, as shown in Fig. 2(a). We assign the patches of a base-grid size to the subgrid boundary, as illustrated by the patches colored in light gray in Fig. 2(a). We place the fine patches *inside* the subgrid. In other words, no fine patches are located on the subgrid interface. Similar to conventional FDTD, the **H** unknowns are placed at the center of each patch. Hence, only base-grid **H** unknowns are located at the subgrid interface. All the fine-patch **H**-unknowns are *inside* the subgrid region. The **E** unknowns are along the edges of the patches. In such a 3-D setting, the e_{cb} fields reside on the coarse edges of the subgrid region, as those marked by blue arrows in Fig. 2(b). The e_{fb} either overlap with e_{cb}, or fall onto the surfaces of the subgrid region, which are shown by red fine edges in Fig. 2(b).

Similar to the algorithm developed in the previous section, between e_{fb} and e_{cb}, we adopt the e_{cb} as the common set of **E** unknowns at the subgrid interface to ensure tangential field continuity. Therefore, our global e vector is the same as that shown in (17), comprising e_{ci}, e_{cb}, and e_{fi}. The e_{fb} is related to e_{cb} by $e_{\text{fb}} = \mathbf{P}_{\text{fc}}e_{\text{cb}}$.

Next, we elaborate how the row vector $\mathbf{S}_e^{(i)}$, and the column vector $\mathbf{S}_h^{(i)}$ are generated for each patch in the proposed algorithm. Since once this is determined, using our single-grid patch-based formulation shown in (7), one can readily obtain an FDTD subgridding algorithm. In the following, we focus on the interface patch, and the fine patches inside the subgrid, as other patches are no different from those in the original FDTD.

For the six base-grid patches residing on the subgrid boundary, their $\mathbf{S}_h^{(i)}$ column vector and $\mathbf{S}_e^{(i)}$ row vector are generated in the same way as how they are generated in the base grid. Hence, there is no special treatment. For the fine patches located inside the subgrid, the $\mathbf{S}_e^{(i)}$ of each patch remains the same as that shown in (3); but in the $\mathbf{S}_h^{(i)}$ of such a fine patch, if there are entries in the rows corresponding to the **E** unknowns falling onto the subgrid boundary, i.e., e_{fb} unknowns, we set these entries to be zero. This is because e_{fb} are not solved, but interpolated from e_{cb}.

The aforementioned procedure does not involve any additional computation as compared to the conventional FDTD, except for the interpolation of e_{fb} from e_{cb}. It results in the following system of equations:

$$\begin{bmatrix} \mathbf{D}_{\epsilon_c} & 0 \\ 0 & \mathbf{D}_{\epsilon_i} \end{bmatrix} \begin{Bmatrix} \ddot{e}_c \\ \ddot{e}_{\text{fi}} \end{Bmatrix} + \begin{bmatrix} \mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c} & 0 \\ \mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,\text{fb}}\mathbf{P}' & \mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,i} \end{bmatrix} \begin{Bmatrix} e_c \\ e_{\text{fi}} \end{Bmatrix} = \begin{Bmatrix} -\dot{j}_c \\ -\dot{j}_i \end{Bmatrix} \quad (45)$$

where $\{e_c\} = \{e_{\text{ci}}\ e_{\text{cb}}\}$ contains all unknowns in the base grid, e_{fi} are inside the subgrid, $\{e\} = \{e_c\ e_{\text{fi}}\}$, subscripts c and i,

respectively, denote the quantities associated with the base grid and the region inside the subgrid. $\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c}$ is the sum of every patch's rank-1 matrix in the base grid, and $\mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,i}$ is the same in the subgrid, but with the $e_{\rm fb}$ unknowns removed. The $[\mathbf{S}_{e,\rm fb}\ \mathbf{S}_{e,i}]$ constitutes the complete $\mathbf{S}_e$ obtained from all fine patches, that is,

$$\mathbf{S}_{e,f} = [\mathbf{S}_{e,\rm fb}\ \mathbf{S}_{e,i}] \tag{46}$$

with $\mathbf{S}_{e,\rm fb}$ corresponding to $e_{\rm fb}$ unknowns, and $\mathbf{S}_{e,i}$ operating on $e_{\rm fi}$. Hence,

$$\mathbf{S}_{e,f}e_f = \mathbf{S}_{e,\rm fb}e_{\rm fb} + \mathbf{S}_{e,i}e_{\rm fi}. \tag{47}$$

The $e_{\rm fb}$ in (45) is replaced by $\mathbf{P}'e_c$, where

$$\mathbf{P}' = [\{0\}_{\#e_{\rm ci}}\ \mathbf{P}_{\rm fc}] \tag{48}$$

such that

$$e_{\rm fb} = \mathbf{P}_{\rm fc}e_{\rm cb}. \tag{49}$$

The numerical system shown in (45) is, in fact, identical to the following two subsystems of equations:

$$\mathbf{D}_{\epsilon_c}\{\ddot{e_c}\} + \mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c}\{e_c\} = -\{\dot{j_c}\} \tag{50}$$

$$\mathbf{D}_{\epsilon_i}\{\ddot{e_i}\} + \mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,i}\{e_i\} = -\{\dot{j_i}\} - \mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,\rm fb}\mathbf{P}'\{e_c\} \tag{51}$$

each of which is SPD since both of $\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c}$ and $\mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,i}$ are SPD. Equation (50) is complete for solving e_c. The term of $\mathbf{S}_{h,i}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,\rm fb}\mathbf{P}'$ in (51) plays a role of interpolating the $e_{\rm cb}$ to obtain the $e_{\rm fb}$, which is then used as a boundary condition for solving the subgrid internal unknowns.

In addition to symmetry, the accuracy of (50) and (51) are guaranteed, since every **H** unknown is generated at the center of an **E** loop, and so is the **E** unknown in the base grid as well as inside the subgrid. The **E** unknowns on the subgrid boundary, $\{e_{\rm fb}\}$, are interpolated from the base-grid solution, the accuracy of which can be controlled to any desired order by the interpolation scheme, without affecting stability. The $\{e_{\rm fb}\}$ can also be solved in the proposed algorithm by obtaining the **H** fields from the base grid adjacent to the $\{e_{\rm fb}\}$, and using them to compute $\{e_{\rm fb}\}$. Again, this does not affect stability. In addition, (50) and (51) allow for the use of different time steps in different regions, since the time step of (50) is solely determined by the $\mathbf{S}_{h,c}\mathbf{D}_{\mu_c^{-1}}\mathbf{S}_{e,c}$, thus the time step of the base grid, while the time step of (51) is determined by the subgrid. At each time step l, (50) is solved to obtain $\{e_c\}^{(l+1)}$ using a large time step Δt_c local to the base grid. The $\{e_c\}$ in each large time interval of Δt_c can be interpolated to obtain its values at instants of Δt_f, the time step allowed by the subgrid. The simulation of (51) can then be carried out using its small time step local to the subgrid.

The entries of $\mathbf{P}_{\rm fc}$ are dependent on the interpolation scheme. Different from many other subgridding methods, any interpolation scheme can be used in the proposed 3-D method without affecting the stability. This is because the $\mathbf{P}_{\rm fc}$-related term is used as the source term independent of the subgrid subsystem to produce fields inside the subgrid, as can be seen from (51). A linear interpolation is used in this paper. As an example, an ith column of $\mathbf{P}_{\rm fc}$ has the following nonzero entries located in the rows correspond to the $e_{\rm fb}$ unknowns interpolated from the ith $e_{\rm cb}$:

$$[\{1\}_n, \{(n-1)/n\}_n, \{(n-2)/n\}_n, \ldots \{1/n\}_n, \{(n-1)/n\}_n, \{(n-2)/n\}_n, \ldots \{1/n\}_n]^T \tag{52}$$

where the subscript n of each subset denotes the number in this subset. For example, the first subset $\{1\}_n$ has n ones, each of which corresponds to one fine edge unknown located on the same edge as the $e_{\rm cb}$ unknown. A single $e_{\rm cb}$ is shared by two patches on the subgrid boundary. The first group of subsets from $\{(n-1)/n\}_n$ to $\{1/n\}_n$ corresponds to the $e_{\rm fb}$ falling onto the first patch and parallel to the $e_{\rm cb}$ unknown. The second group of subsets from $\{(n-1)/n\}_n$ to $\{1/n\}_n$ corresponds to those on the second patch.

B. Relationship With the Operator Developed in Section III

The subgridding algorithm algebraically derived in Section III is, in fact, general for both 2- and 3-D problems. The key result of this algorithm is given by (29). Depending on the choice of **P**, the resultant numerical scheme can be different. We can show the new 3-D algorithm developed in the previous section is (29) with a new 3-D interpolation **P** as follows.

To align with the operator developed in Section III, we again will place all fine patches on the subgrid interface. There are both $e_{\rm fb}$ unknowns distributed along the edges, which are denoted by $e_{{\rm fb},e}$; and $e_{\rm fb}$ unknowns residing on the faces of the subgrid interface, which are denoted by $e_{{\rm fb},f}$. Both of them are interpolated from $e_{\rm cb}$, that is,

$$\{e_{\rm fb}\} = [e_{{\rm fb},e}\ e_{{\rm fb},f}] = \mathbf{P}_{\rm fc}e_{\rm cb} \tag{53}$$

and $\mathbf{P}_{\rm fc}$ has entries shown in (52). The $\mathbf{P}_{\rm fc}$ can further be split as

$$\mathbf{P}_{\rm fc} = \begin{bmatrix} \mathbf{P}_{{\rm fc},e} \\ \mathbf{P}_{{\rm fc},f} \end{bmatrix} \tag{54}$$

where $\mathbf{P}_{{\rm fc},e}e_{\rm cb}$ produces edge-based $e_{{\rm fb},e}$, and $\mathbf{P}_{{\rm fc},f}$ when operating on $e_{\rm cb}$ yields face $e_{{\rm fb},f}$. Each column of $\mathbf{P}_{{\rm fc},e}$ is an all-one vector, the same as (30).

When forming the subgrid system of equations shown in (21), only $e_{{\rm fb},e}$ are used, since only their equations need to be added upon the equation of $e_{\rm cb}$ generated from the base grid to complete $e_{\rm cb}$'s solution. Hence, $\{e_{\rm sub}\}$ of (21) becomes

$$\{e_{\rm sub}\} = \{e_{\rm ci}\ e_{{\rm fb},e}\ e_{\rm fi}\}^T. \tag{55}$$

Using (54), $\{e_{\rm sub}\}$ can be written as

$$\{e_{\rm sub}\} = \mathbf{P}_e\{e\} \tag{56}$$

where global mapping matrix $\mathbf{P}_e$ is

$$\mathbf{P}_e = \begin{bmatrix} (\mathbf{I})_{\#e_{\rm ci}\times\#e_{\rm ci}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{P}_{{\rm fc},e})_{\#e_{{\rm fb},e}\times\#e_{\rm cb}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (\mathbf{I})_{\#e_{\rm fi}\times\#e_{\rm fi}} \end{bmatrix}. \tag{57}$$

The subscript e of $\mathbf{P}_e$ is used to highlight the fact that only edge-based $e_{{\rm fb},e}$, and hence $\mathbf{P}_{{\rm fc},e}$ part of $\mathbf{P}_{\rm fc}$ is involved in (57).

Because $\{e_{\text{sub}}\}$ only includes $e_{\text{fb},e}$, the $\mathbf{S}_{h,f}$ in the right-hand side of (21) does not include rows corresponding to $e_{\text{fb},f}$. Hence, when generating a column vector $\mathbf{S}_h^{(i)}$ in the fine patch, if the row entries correspond to $e_{\text{fb},f}$, they are excluded. As a result, (25) becomes

$$\mathbf{D}_{\epsilon,f}\mathbf{P}_e\{\ddot{e}\}_- = -\mathbf{S}_{h,f0}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\{e\} \tag{58}$$

where $\mathbf{S}_{h,f0}$ denotes $\mathbf{S}_{h,f}$ with its rows corresponding to $e_{\text{fb},f}$ removed, the right-hand side $\mathbf{P}$ has the complete $\mathbf{P}_{\text{fc}}$ to obtain e_{fb}, as shown in the following:

$$\mathbf{P} = \begin{bmatrix} (\mathbf{I})_{\#e_{\text{ci}}\times\#e_{\text{ci}}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\mathbf{P}_{\text{fc}})_{\#e_{\text{fb}}\times\#e_{\text{cb}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (\mathbf{I})_{\#e_{\text{fi}}\times\#e_{\text{fi}}} \end{bmatrix}. \tag{59}$$

Since among all e_{fb} unknowns, only $e_{\text{fb},e}$ are computed in (58), not all $e_{\text{fb},f}$ unknowns are involved in $\mathbf{P}\{e\}$ in the right-hand side of (58). Only those $e_{\text{fb},f}$ in the fine patches having $e_{\text{fb},e}$, i.e., immediately adjacent to e_{cb} are involved.

From (58), we obtain

$$\mathbf{D}_{\epsilon}\{\ddot{e}\}_- = -\left(\mathbf{P}_e^T\mathbf{P}_e\right)^{-1}\mathbf{P}_e^T\mathbf{S}_{h,f0}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,f}\mathbf{P}\{e\} \tag{60}$$

where $(\mathbf{P}_e^T\mathbf{P}_e)^{-1}$ has the same form as that shown in (31). The above again means using the average magnetic field across all fine cells immediately adjacent to e_{cb} to perform the curl of $\mathbf{H}$ operation from the subgrid side. The other half of the curl of $\mathbf{H}$ operation is completed by equations generated from the base grid. By writing out the right-hand side of (60) explicitly, and using the same $\mathbf{P}_{\text{fc}}$ shown in (52), we find e_{cb} is, in fact, generated in the following way:

$$\mathbf{D}_{\epsilon}\{\ddot{e_{\text{cb}}}\}_- = -\mathbf{S}_{h,0}\mathbf{D}_{\mu_f^{-1}}\mathbf{S}_{e,0}\{e\} \tag{61}$$

where $\mathbf{S}_{h,0}$ and $\mathbf{S}_{e,0}$ are the same as those formulated on the base-grid patches placed on the subgrid interface. Hence, this is the same as the new 3-D algorithm described in Section V-A.

VI. Numerical Results

We have simulated a variety of 2- and 3-D examples to validate the proposed subgridding algorithms. All of the 2-D examples are simulated by using the formulations given in Section IV; and the 3-D examples are analyzed by using the algorithm described in Section V. The CPU time reported in this paper is the total CPU time that includes the time cost of every step.

A. 2-D Free-Space Wave Propagation Having a Subgrid

The first example is a free-space wave propagation problem in a 2-D region of size 0.5 m by 0.5 m. The grid details are shown in Fig. 3(a). The coarse grid size is $L_c = 0.1$ m. The subgrid marked in blue is located in the center having a grid ratio n from 2, 5, 10, to 100. The $\mathbf{E}^{\text{inc}}$ is $\hat{y}2(t - t_0 - x/c)e^{-(t-t_0-x/c)^2/\tau^2}$ with $c = 3 \times 10^8$ m/s, $\tau = 2 \times 10^{-8}$ s and $t_0 = 4\tau$. All the boundaries are terminated by an exact absorbing boundary condition. To evaluate the accuracy of the proposed method, the entire solution error at each time step as compared to the analytical solution, $\|\{e\} - \{e\}_{\text{anal}}\|/\|\{e\}_{\text{anal}}\|$ with $\{e\}$ containing all N_e unknowns, is plotted versus time in Fig. 3(b) for various grid ratios. Excellent accuracy is observed for all the grid ratios. The large error shown in early and late time is because fields at that time are zero, and the theoretical value of the error is infinity. To demonstrate the efficiency of the proposed FDTD subgridding method, we also simulate the same problem using the conventional FDTD method in a uniform mesh. The simulation parameters are summarized in Table I. Good speedup is achieved over the conventional FDTD without using a subgrid.

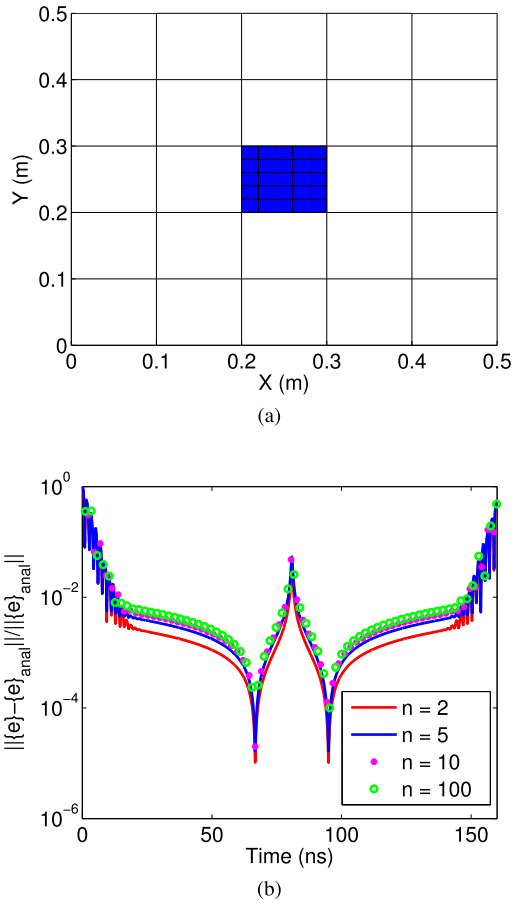


Fig. 3. Simulation of a 2-D wave propagation problem. (a) Grid with a subgrid. (b) Entire solution error versus time for different grid ratios.

TABLE I
Performance Comparison for Different Grid Ratios

Grid Ratio		2	5	10	100
Time Step (s)		1.4e-10	4.9e-11	2.3e-11	2.3e-12
Num. of E	FDTD	220	1300	5100	501,000
	This Method	68	116	276	20256
Time (s)	FDTD	0.12	0.68	3.92	11237.47
	This Method	0.04	0.24	0.47	169.89
Speedup		3	2.83	8.34	66.14

B. 2-D PEC Cavity With Conducting Fins

The second example is a PEC cavity with two conducting fins separated by a thin gap, as shown in Fig. 4(a). The fins are of finite conductivity 5.8×10^7 S/m. A current source is launched at the middle of the fin gap vertically, the pulse

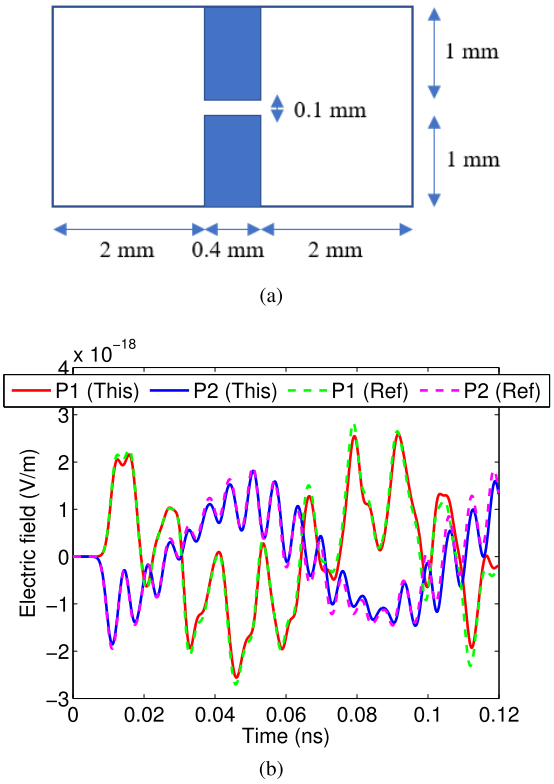


Fig. 4. Cavity with lossy conductors and fine features. (a) Structure. (b) Simulated electric fields.

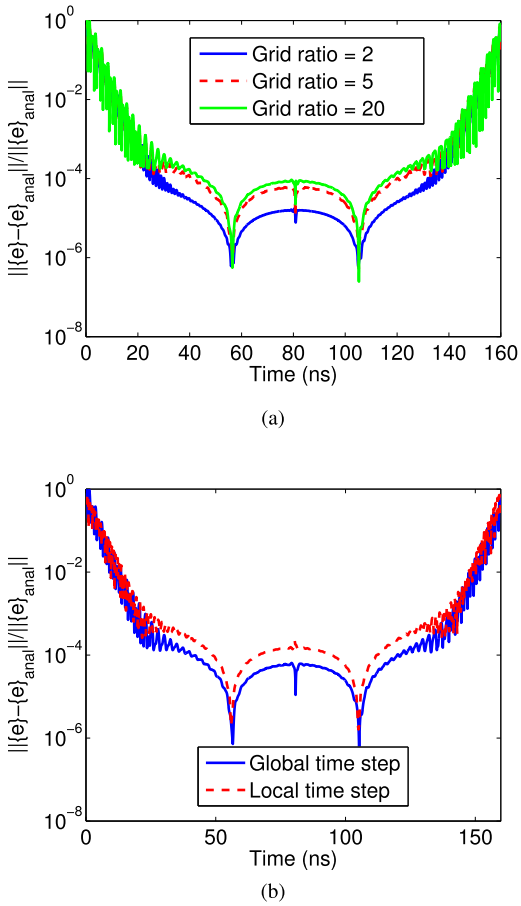


Fig. 5. Simulation of a 3-D wave propagation problem. (a) Entire solution error versus time with different grid ratios, and a global small time step. (b) Entire solution error versus time with local time stepping as compared to using a global small time step across the grid.

of which is $-\tau^2 e^{-(t-t_0)^2/\tau^2}$ with $\tau = 2 \times 10^{-12}$ s and $t_0 = 4\tau$. The region between the two fins is discretized into subgrids. The coarse grid size is 0.1 m, while the subgrid size is 0.05 m. The number of **E** unknowns used in the proposed method is 1932, as compared to the 7532 **E** unknowns used in the FDTD without subgridding. The traditional FDTD method takes 1.06 s to finish the simulation while the proposed method only takes 0.24 s, so the speedup of the proposed method is 4.4. As can be seen in Fig. 4(b), the fields simulated from this method agree very well with the reference FDTD solution generated from a uniform grid. The two points simulated are at (1, 1.05) and (2.85, 1.5) mm, respectively.

C. 3-D Free-Space Wave Propagation Having a Subgrid

Next, we simulate a free-space wave propagation problem in a 3-D cube, the side length of which is 0.5 m. Along all directions, the base grid step L_c is 0.1 m, resulting in 125 coarse cells. The coarse cell at the center is further subdivided into fine cells with grid ratio n, therefore the fine grid size L_f is $0.1/n$ m. The n is chosen to be 2, 5, and 20, respectively. The incident field is $\mathbf{E}^{\text{inc}} = \hat{y}2(t - t_0 - x/c)e^{-(t-t_0-x/c)^2/\tau^2}$ with $c = 3 \times 10^8$ m/s, $\tau = 2 \times 10^{-8}$ s and $t_0 = 4\tau$. An exact absorbing boundary condition is applied to truncate the outermost boundary. The time step required by fine cells in the subgrid is 4.0×10^{-11} s. To examine the solution accuracy in the entire computational domain, in Fig. 5(a), we assess the entire solution error measured by $\|\{e\} - \{e\}_{\text{anal}}\|/\|\{e\}_{\text{anal}}\|$ when grid ratio is 2, 5, and 20, respectively, where $\{e\}$ consists of all **E** unknowns obtained from the proposed FDTD subgridding method, while $\{e\}_{\text{anal}}$ is from the analytical result. As can be seen clearly, the proposed method is accurate at all points, and across the whole time window simulated.

The proposed method also allows the usage of different time steps in different domains. To verify this feature, we re-simulate the same example using $\Delta t_f = 4.0 \times 10^{-11}$ s in the subgrid region, and $\Delta t_c = 2.0 \times 10^{-10}$ s in the base grid for grid ratio $n = 5$. The entire solution error is plotted in Fig. 5(b). For comparison, the solution error is also plotted for the case when the small time step Δt_f is used in the entire computational domain. As can be seen clearly, using local time stepping, the proposed method equally produces accurate and stable results. The accuracy is only degraded slightly, as compared to using a global small time step all over the grid. The simulation using a global time step takes 1.0543 s to finish, while the simulation using different time steps in different grids only takes 0.2782 s to finish.

D. 3-D PEC Cavity With a Lossy Conducting Object

We then simulate a 3-D cavity excited by a current source. The cavity is 1-cm long in all directions and its six faces are all terminated by a PEC boundary condition. The coarse grid size along each direction is 1 mm except for the small cube

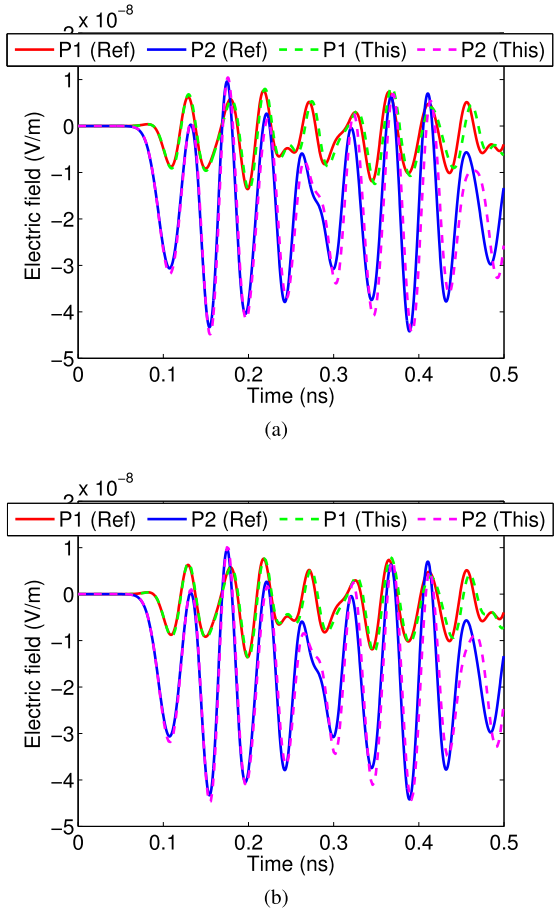


Fig. 6. Simulation of a 3-D PEC cavity with fine conducting features discretized into subgrids. (a) Electric fields at two observation points using a global time stepping. (b) Electric fields at two observation points using a local time stepping.

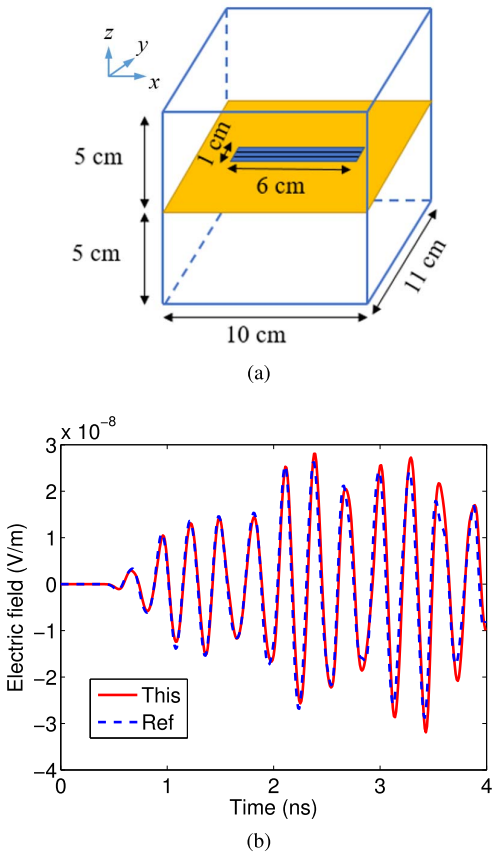


Fig. 7. Simulation of an EMI example. (a) Illustration of the structure. (b) Simulated electric field in comparison with reference solution.

centered at (4.5, 4.5, and 4.5) mm. This cube is 1-mm long in all directions and filled with a conductive material whose conductivity is 5.7×10^7 S/m. This cube is further subdivided into fine mesh whose grid size is 0.2 mm. Such a subgridding mesh results in 3870 **E** unknowns. A current probe is excited at (2, 2, and 1.5) mm. The current is a Gaussian pulse whose waveform is $\mathbf{I} = \hat{z}\tau e^{-(t-t_0)^2/\tau^2}$ with $\tau = 2 \times 10^{-11}$ s and $t_0 = 4\tau$. As the reference, we also simulate the same problem using the conventional FDTD method with a uniform fine mesh. The total number of **E** unknowns in this uniform fine mesh is 390, 150. The time step allowed by the conventional FDTD method is $\Delta t = 3.8 \times 10^{-13}$ s. In Fig. 6(a), the electric fields sampled at point 1 (8, 8, and 7.5) mm and point 2 (4, 4, and 9.5) mm are plotted in comparison with reference solutions. Overall, the accuracy of the simulated fields is very good. The small discrepancy in the waveform can be attributed to the excitation of the cavity spurious solutions after the wave reaches the PEC wall. These spurious solutions (source-free solutions of the cavity) that are captured numerically are different in different grids used for discretization. As for the CPU time, the proposed FDTD subgridding method only takes 0.24 s to finish the entire simulation, while the conventional FDTD method requires 68.79 s, thus a significant speedup is achieved. This efficiency is further improved when using a local time stepping. We use a large time step $\Delta t_c = 1.9 \times 10^{-12}$ s in the base grid region, while the time step in the subgrid region remains the same. The electric fields sampled at the same observation points are plotted in Fig. 6(b). Obviously, the solution accuracy is not sacrificed. Meanwhile, the CPU time is reduced to 0.11 s.

E. Electromagnetic Interference Example

In this example, we simulate an electromagnetic interference (EMI) example as illustrated in Fig. 7(a). The structure is a box of size $10 \times 11 \times 10$ cm truncated by perfect electric boundary conditions all around. In the center, there is a PEC sheet with a thin slot. The slot is 1-cm wide and 6-cm long. The coarse grid size along each direction is 1 cm, while the subgrid size along y-direction for the slot is $1/3$ cm. This structure is excited by a current source of $\mathbf{I} = \hat{y} \exp -(t - t_0/\tau)^2$ located at (5, 1.5, and 2) cm, where $\tau = 5 \times 10^{-11}$ s and $t_0 = 4\tau$. There are 3971 unknowns in this mesh. As the reference, we use a uniform fine grid of size $1/3$ cm along y-direction and simulate the example using the traditional FDTD method. The number of unknowns used in the traditional FDTD is 11/,473. To finish the simulation, the proposed subgridding algorithm takes 0.074 s while the traditional FDTD method takes 0.275 s, so the CPU time speedup is 3.72. To verify the accuracy, we sample the electric field at the point (5, 1.5, and 8) cm in

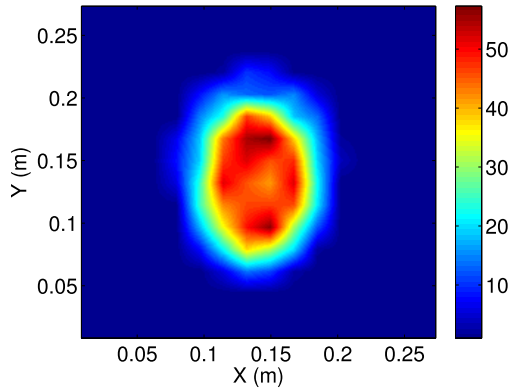


Fig. 8. Relative permittivity distribution in a cross section of the phantom head at $z = 2.8$ cm.

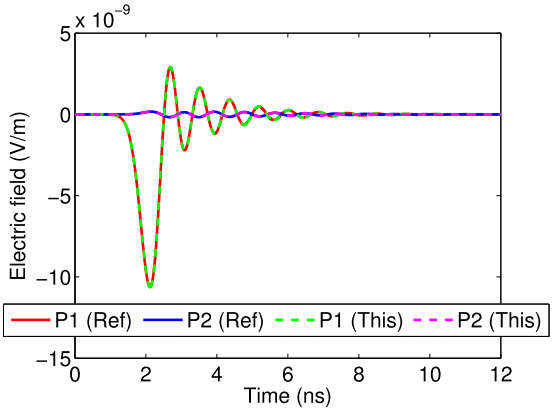


Fig. 9. Simulated electric fields at two observation points in comparison with reference FDTD solutions when local time stepping is used.

both the methods and plot it in Fig. 7(b). Overall, the accuracy is very good across the whole time window.

F. Inhomogeneous 3-D Phantom Head Beside a Wire Antenna

The last example we study is a large-scale phantom head beside a wire antenna, which involves many inhomogeneous materials. The size of the phantom head is 28.16 cm × 28.16 cm × 17.92 cm. The permittivity distribution of the head at $z = 2.8$ cm is shown in Fig. 8. All the boundaries are truncated by PMC. The wire antenna is located at (3.52, 3.52, and 2.52) cm, the current at which has a pulse waveform of $\mathbf{I} = 2(t-t_0)e^{-(t-t_0)^2/\tau^2}$ with $\tau = 5.0\times10^{-10}$ s and $t_0 = 4\tau$. The coarse step size along x-, y-, z-directions is 4.4, 4.4, and 5.6 mm, respectively. To capture fine tissues, the coarse cell centered at (14.3, 14.3, and 9.24) cm are subdivided into fine cells in all directions with grid ratio $n = 4$, meaning the fine grid size along x-, y-, z-directions is 1.1, 1.1, and 1.4 mm, respectively. As a result, the total number of $\mathbf{E}$ unknowns in this subgridding mesh is 409 868. In conventional FDTD, if fine grids are used everywhere, it would require 25 428 608 $\mathbf{E}$ unknowns. Due to the existence of fine grids, the conventional FDTD method must use a time step of 2.2×10^{-12} s across the whole grid to ensure stability. In contrast, this method allows to use a larger time step of $\Delta t_c = 8.8 \times 10^{-12}$ s in the base grid. In Fig. 9, the electric fields at two observation points whose locations are (3.52, 3.52, and 15.96) cm and (24.64, 3.52, and 15.96) cm are plotted in comparison with the reference FDTD solutions. It is clear that the result from the proposed method agrees well with the reference result. Since the conventional FDTD method requires a uniform fine grid which has many more unknowns than that of the proposed method, it takes the conventional FDTD method 19 222.16 s to finish the simulation. In contrast, the proposed subgridding method only costs 105.03 s.

VII. Conclusion

In this paper, we develop a systematic approach to algebraically derive FDTD subgridding operators that are accurate and SPD by construction. This is in contrast to conventional methods where the subgridding scheme is first developed, whose stability analysis is then attempted. As also shown in this paper, as long as the system matrix becomes unsymmetrical, in general, it can have complex eigenvalues, which would make an explicit marching absolutely unstable. In this paper, starting from a patch-based single-grid matrix representation of the FDTD, we first study how subdomain equations are assembled in FDTD to build a global system of equations. Based on the findings of this study, an algebraic approach is developed to systematically derive SPD subgridding operators. Both 2- and 3-D subgridding algorithms are developed with guaranteed stability and accuracy. They show no limitation on grid ratio, and no requirement on the special arrangement of the subgrids. Local time stepping is also realized in this paper. Extensive 2- and 3-D simulations with various grid ratios and inhomogeneous materials as well as fine features have demonstrated the accuracy, stability, and efficiency of the proposed new subgridding algorithms.

References

[1] S. S. Zivanovic, K. S. Yee, and K. K. Mei, "A subgridding method for the time-domain finite-difference method to solve Maxwell's equations," *IEEE Trans. Microw. Theory Techn.*, vol. 39, no. 3, pp. 471–479, Mar. 1991.

[2] D. T. Prescott and N. V. Shuley, "A method for incorporating different sized cells into the finite-difference time-domain analysis technique," *IEEE Microw. Guided Wave Lett.*, vol. 2, no. 11, pp. 434–436, Nov. 1992.

[3] M. J. White, M. F. Iskander, and Z. Huang, "Development of a multigrid FDTD code for three-dimensional applications," *IEEE Trans. Antennas Propag.*, vol. 45, no. 10, pp. 1512–1517, Oct. 1997.

[4] K. Xiao, D. J. Pommerenke, and J. L. Drewniak, "A three-dimensional FDTD subgridding algorithm with separated temporal and spatial interfaces and related stability analysis," *IEEE Trans. Antennas Propag.*, vol. 55, no. 7, pp. 1981–1990, Jul. 2007.

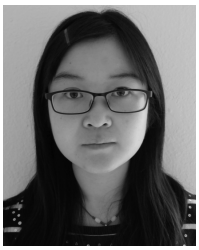
[5] J. Yan and D. Jiao, "Accurate and stable matrix-free time-domain method in 3-D unstructured meshes for general electromagnetic analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 12, pp. 4201–4214, Dec. 2015.

[6] M. W. Chevalier, R. J. Luebbers, and V. P. Cable, "FDTD local grid with material traverse," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 411–421, Mar. 1997.

[7] M. Okoniewski, E. Okoniewska, and M. A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 422–429, Mar. 1997.

[8] M. J. White, Z. Yun, and M. F. Iskander, "A new 3D FDTD multigrid technique with dielectric traverse capabilities," *IEEE Trans. Microw. Theory Techn.*, vol. 49, no. 3, pp. 422–430, Mar. 2001.

[9] P. Thoma and T. Weiland, "A consistent subgridding scheme for the finite difference time domain method," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 9, no. 5, pp. 359–374, Sep. 1996.
[10] O. Podebrad, M. Clemens, and T. Weiland, "New flexible subgridding scheme for the finite integration technique," *IEEE Trans. Magn.*, vol. 39, no. 3, pp. 1662–1665, May 2003.
[11] L. Kulas and M. Mrozowski, "Reciprocity principle for stable subgridding in the finite difference time domain method," in *Proc. Int. Conf. Comput. Tool (EUROCON)*, Sep. 2007, pp. 106–111.
[12] J. Yan and D. Jiao, "A symmetric positive semi-definite FDTD subgridding algorithm for arbitrary grid ratios with uncompromised accuracy," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2017, pp. 1–4.
[13] J. Yan and D. Jiao, "Explicit and unconditionally stable FDTD method without eigenvalue solutions," in *IEEE MTT-S Int. Microw. Symp. Dig.*, May 2016, pp. 1–4.
[14] J. Yan and D. Jiao, "Fast explicit and unconditionally stable FDTD method for electromagnetic analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 8, pp. 2698–2710, Aug. 2017.
[15] M. Gaffar and D. Jiao, "An explicit and unconditionally stable FDTD method for electromagnetic analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 11, pp. 2538–2550, Nov. 2014.



Jin Yan (GS'13–M'17) received the B.S. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2012, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2016.

She has been a Software Engineer with the Platform Engineering Group, Intel Corporation, Hillsboro, OR, USA, since 2016. Her current research interests include signal and power integrity, computational electromagnetics, high-performance VLSI CAD, and fast and high-capacity numerical methods.

Dr. Yan was a recipient of an Honorable Mention Award of the IEEE International Symposium on Antennas and Propagation in 2015 and the Best Student Paper Award Finalist from the IEEE MTT-S International Microwave Symposium in 2016.

Dan Jiao (M'02–SM'06–F'16) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2001.

She was with the Technology Computer-Aided Design (CAD) Division, Intel Corporation, Hillsboro, OR, USA, in 2005, as a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. She then joined Purdue University, West Lafayette, IN, USA, as an Assistant Professor with the School of Electrical and Computer Engineering, where she is currently a Professor. She has authored 3 book chapters and over 260 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics, high-frequency digital, analog, mixed-signal, and RF integrated circuit design and analysis, high-performance VLSI CAD, modeling of microscale and nanoscale circuits, applied electromagnetics, fast and high-capacity numerical methods, fast time-domain analysis, scattering and antenna analysis, RF, microwave, millimeter-wave circuits, wireless communication, and bioelectromagnetics.

Dr. Jiao has served as the reviewer for many IEEE journals and conferences. She is an Associate Editor of the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING, AND MANUFACTURING TECHNOLOGY. She was a recipient of the 2013 S. A. Schelkunoff Prize Paper Award of the IEEE Antennas and Propagation Society, which recognizes the Best Paper published in the IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION during the previous year. She was among the 21 women faculty selected across the country as the 2014–2015 Fellow of Executive Leadership in Academic Technology and Engineering at Drexel, a national leadership program for women in the academic STEM fields. She has been named a University Faculty Scholar by Purdue University since 2013. She was among the 85 engineers selected throughout the nation for the National Academy of Engineerings 2011 U.S. Frontiers of Engineering Symposium. She was a recipient of the 2010 Ruth and Joel Spira Outstanding Teaching Award, the 2008 National Science Foundation CAREER Award, the 2006 Jack and Cathie Kozik Faculty Startup Award (which recognizes an outstanding new faculty member of the School of Electrical and Computer Engineering, Purdue University), the 2006 Office of Naval Research (ONR) Award under the Young Investigator Program, the 2004 Best Paper Award presented at the Intel Corporation's annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits, the 2003 Intel Corporation's Logic Technology Development (LTD) Divisional Achievement Award, the Intel Corporation's Technology CAD Divisional Achievement Award, the 2002 Intel Corporation's Components Research the Intel Hero Award (Intel-wide she was the 10th recipient), the Intel Corporation's LTD Team Quality Award, and the 2000 Raj Mittra Outstanding Research Award presented by the University of Illinois at Urbana–Champaign.