Accurate and Stable Matrix-Free Time-Domain Method in 3-D Unstructured Meshes for General Electromagnetic Analysis

Jin Yan, *Student Member, IEEE*, and Dan Jiao, *Senior Member, IEEE*

***Abstract*—We develop a new time-domain method that is naturally matrix free, i.e., requiring no matrix solution, regardless of whether the discretization is a structured grid or an unstructured mesh. Its matrix-free property, manifested by a naturally diagonal mass matrix, is independent of the element shape used for discretization and its implementation is straightforward. No dual mesh, interpolation, projection, and mass lumping are required. Furthermore, we show that such a capability can be achieved with conventional vector basis functions without any need for modifying them. Moreover, a time-marching scheme is developed to ensure the stability for simulating an unsymmetrical numerical system whose eigenvalues can be complex-valued and even negative, while preserving the matrix-free merit of the proposed method. Extensive numerical experiments have been carried out on a variety of unstructured triangular, tetrahedral, triangular prism element, and mixed-element meshes. Correlations with analytical solutions and the results obtained from the time-domain finite-element method, at all points in the computational domain and across all time instants, have validated the accuracy, matrix-free property, stability, and generality of the proposed method.**

***Index Terms*—Electromagnetic analysis, finite-difference time domain (FDTD) methods, matrix-free methods, time-domain finite-element methods, time-domain methods, unstructured mesh.**

I. Introduction

MANY engineering challenges demand an efficient computational solution of large-scale problems. If a computational method can be made matrix free, i.e., free of matrix solutions, then it has a potential of solving very large scale problems. Among existing computational electromagnetic methods, the explicit finite-difference time-domain (FDTD) method [1], [2] is free of matrix solutions. However, it requires a structured orthogonal grid for space discretization. To overcome this limitation, many nonorthogonal FDTD methods have been developed such as the curvilinear FDTD [3]–[5], contour and conformal FDTD [6]–[8], discrete surface integral (DSI) methods [9], generalized Yee-algorithms [10]–[15], and the Finite Integration Technique with affine theories [16]. Needless to say, they have significantly advanced the capability of the original FDTD method in handling unstructured meshes.

In existing nonorthogonal FDTD methods, a dual mesh is generally required. The dual mesh needs to satisfy a certain relationship with the primary mesh. Such a dual mesh may not exist in an unstructured mesh. For cases where the dual mesh exists, the accuracy of many nonorthogonal FDTD methods can still be limited. This is because in these methods, the field unknowns are placed along the edges of either the primary mesh or the dual mesh, and are assumed to be constant along the edges. Restricted by such a representation of the fields, one can only obtain the dual field accurately (second-order accurate) at the center point of the loop of the primary field, and along the direction normal to the loop area. Elsewhere and/or along other directions, the accuracy of the dual field cannot be ensured. However, the points and directions, where the dual fields can be accurately obtained, are not coincident with the points and directions of the dual fields located on the dual mesh, in an unstructured mesh. Actually, the only mesh that can align the two is an orthogonal grid, which is used by the traditional FDTD method. As a result, the desired dual fields have to be obtained by interpolations and projections, the accuracy of which is difficult to control in an arbitrary unstructured mesh. It is observed that many interpolation and projection schemes lack a theoretical error bound. The same is true to the primary fields obtained from the dual fields.

In addition to accuracy, stability is another concern since the curl operation on $\mathbf{E}$ is, in general, not reciprocal to that on $\mathbf{H}$ in existing methods developed for irregular meshes. It can be proved that such a nonreciprocal operation can result in complex-valued or negative eigenvalues in the underlying numerical system. They make a traditional explicit time-marching absolutely unstable. This fact was also made clear in [15]. As a consequence, it remains a research problem how to ensure both accuracy and stability of an FDTD-like method in an unstructured mesh.

The finite-element method in time domain (TDFEM) [17] has no difficulty in handling arbitrarily shaped irregular meshes, but it requires the solution of a mass matrix, thus not being matrix-free in nature. The mass-lumping has been used to diagonalize the mass matrix in TDFEM, and also finite integration technique [16]. But it requires well-shaped elements to be accurate [18]. In addition to mass lumping, orthogonal vector basis functions have been developed to render the mass matrix diagonal [19], [20]. These bases are element-shape dependent. They

Manuscript received June 15, 2015; revised September 05, 2015; accepted October 19, 2015. Date of publication November 17, 2015; date of current version December 02, 2015. This work was supported in part by the NSF under Grant 1065318, and by DARPA under Grant HR0011-14-1-0057. This paper is an expanded version from the IEEE MTT-S International Microwave Symposium, Phoenix, AZ, USA, May 17–22, 2015.

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.2015.2495257

0018-9480 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

also rely on an approximate integration to make the mass matrix diagonal. In recent years, Discontinuous Galerkin time-domain methods [21], [22] have been developed, which only involve the solution of local matrices of small size. However, this is achieved by *not* enforcing the tangential continuity of the fields across the element interface at the same time instant. Certainly, an accurate result would still have to satisfy the continuity conditions of the fields. Not satisfying them has implications in either accuracy or efficiency. For example, it is observed that a Discontinuous Galerkin time-domain method typically requires a time step much smaller than that of a traditional explicit time-domain method for accurate transient analysis. Recently, a new time-domain method is developed in [23], which requires no matrix solution regardless of whether the discretization is a structured grid or an unstructured mesh. Since the curl operation on $\mathbf{E}$ and that of $\mathbf{H}$ are enforced to be reciprocal of each other in [23], although the stability is guaranteed for an arbitrary unstructured mesh, the accuracy remains to be a strong function of mesh quality.

In this paper, we develop an accurate and stable matrix-free time-domain method that is independent of the element shape used for discretization. The tangential continuity of the fields is satisfied across the element interface at each time instant. No dual mesh, interpolation, projection, and mass-lumping are needed. The accuracy and stability are both guaranteed for an arbitrary unstructured mesh. This method is also made very easy to implement. In addition, in a structured grid and with zeroth-order vector bases, the proposed method reduces exactly to the FDTD. The basic idea of this paper was outlined in [24], where 2-D formulations are provided, and modified higher-order bases are developed to achieve a matrix-free method. In this paper, we present 3-D formulations of [24] for general electromagnetic analysis. We also show the proposed matrix-free method can be formulated without modifying the traditional vector basis functions. In addition, a comprehensive analysis is conducted on the accuracy and stability of the proposed method. Numerical results on various highly unstructured triangular, tetrahedral, triangular prism meshes as well as meshes with mixed-elements are presented to demonstrate the accuracy, matrix-free property, and generality of the proposed method.

II. Proposed Framework for Creating a Matrix-Free Time-Domain Method

In this section, we present a general framework for creating a matrix-free time-domain method independent of the shape of the elements used for discretization. We separate the presentation of the framework from that of the detailed formulations (to be given in next section) because the formulation corresponding to the proposed framework is not unique. Under the proposed framework, we can develop different formulations to achieve a matrix-free time-domain method.

Consider a general electromagnetic problem involving arbitrarily shaped geometries and materials. For such a problem, an unstructured mesh with *arbitrarily shaped* elements is more accurate and efficient for use, as compared to an orthogonal grid. The elements do not have to be of the same type. They can be a mix of different types of elements such as tetrahedral, triangular prism, and brick elements. Starting from the differential form of Faraday's law and Ampere's law

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \tag{1}$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E} + \mathbf{J} \tag{2}$$

we pursue a discretization of the two equations in time domain, which can yield a numerical system free of matrix solutions independent of the element shape used for discretization.

A. Discretization of Faraday's Law

To discretize Faraday's law, we propose to expand the electric field $\mathbf{E}$ in each element by a set of vector bases $\mathbf{N}_j$ $(j = 1, 2, \ldots, m)$ as the following:

$$\mathbf{E} = \sum_{j=1}^{m} u_j \mathbf{N}_j \tag{3}$$

where u_j is the unknown coefficient of the jth vector basis $\mathbf{N}_j$, and m is the number of vector bases in each element. The degrees of freedom of the vector bases $\mathbf{N}$ are defined not only on the faces of the element but also inside the element. Such a choice of vector bases permits accurate generation of the other field unknown at any point along an arbitrary direction, without a need for interpolation and projection.

Substituting the expansion of $\mathbf{E}$ into (1), computing $\mathbf{H}$ at N_h points $\mathbf{r}_{hi}$ $(i = 1, 2, \ldots, N_h)$, and then taking the dot product of the resultant with unit vector $\hat{h}_i$ at each point respectively, we obtain

$$\hat{h}_i \cdot \sum u_j \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}) = -\hat{h}_i \cdot \mu(\mathbf{r}_{hi}) \frac{\partial \mathbf{H}(\mathbf{r}_{hi})}{\partial t}, \quad (i = 1, 2, \ldots, N_h) \tag{4}$$

which can be compactly written into the following linear system of equations:

$$\mathbf{S}_e\{u\} = -\text{diag}\left(\{\mu\}\right) \frac{\partial \{h\}}{\partial t} \tag{5}$$

where $\text{diag}(\{\mu\})$ is a diagonal matrix of the permeability, $\{h\}$ is a global vector of length N_h whose ith entry is

$$h_i = \mathbf{H}(\mathbf{r}_{hi}) \cdot \hat{h}_i \tag{6}$$

and $\mathbf{S}_e$ is a sparse matrix, the nonzero entries of which are

$$\mathbf{S}_{e,ij} = \hat{h}_i \cdot \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}) \tag{7}$$

where i denotes the global index of the $\mathbf{H}$-point, and j is the global index of the $\mathbf{E}$'s vector basis function. Let N_e be the total number of vector bases used to expand $\mathbf{E}$. The $\mathbf{S}_e$ is of size $N_h \times N_e$. During the procedure of constructing $\mathbf{S}_e$, the tangential continuity of $\mathbf{E}$ is enforced since the tangential electric fields at the element interface are uniquely defined in global vector $\{u\}$, and shared in common by all elements.

B. Discretization of Ampere's Law

To discretize Ampere's law, we apply it at $\mathbf{r}_{ei}$ $(i = 1, 2, \ldots, N_e)$ points, and then take the dot product of the resultant with unit vector $\hat{e}_i$ at each point, where $\mathbf{r}_{ei}$ and $\hat{e}_i$ are

associated with the degrees of freedom of the vector bases used in (3). We obtain

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = \epsilon(\mathbf{r}_{ei})\frac{\partial e_i}{\partial t} + \sigma(\mathbf{r}_{ei})e_i + \hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei}) \quad (8)$$

where

$$e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i \quad (9)$$

which is $\mathbf{E}$ at point $\mathbf{r}_{ei}$ along the $\hat{e}_i$ direction. The $\hat{e}_i \cdot \nabla \times \mathbf{H}$ at point $\mathbf{r}_{ei}$ in (8) is generated by using the $\mathbf{H}$ fields [obtained from (5)] encircling e_i. For example, if e_i is located at an element interface, the $\mathbf{H}$ fields used to generate it are sampled across the elements sharing e_i. A detailed formulation with guaranteed accuracy will be given in next section. As a result, we obtain the following discretization of Ampere's law

$$\mathbf{S}_h\{h\} = \mathrm{diag}\left(\{\epsilon\}\right)\frac{\partial\{e\}}{\partial t} + \mathrm{diag}\left(\{\sigma\}\right)\{e\} + \{j\} \quad (10)$$

where $\mathbf{S}_h$ is a sparse matrix of size $N_e \times N_h$, and $\mathbf{S}_h\{h\}$ denotes the discretized $\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei})$ $(i = 1, 2, \ldots, N_e)$ operation, the ith entry of $\{j\}$ is $\hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei})$, and the $\mathrm{diag}(\{\epsilon\})$ and $\mathrm{diag}(\{\sigma\})$ are the diagonal matrices of permittivity, and conductivity respectively.

C. Connecting Ampere's Law to Faraday's Law

In order to connect (10) to (5), we need to find the relationship between $\{e\}$ and $\{u\}$. In [24], by making a minor modification of the traditional vector bases, we make $\{u\} = \{e\}$. In this work, we show the traditional vector bases can also be kept as they are without any need for modification. In this case, we can find an analytical relationship between $\{e\}$ and $\{u\}$ as $\{u\} = \mathbf{Q}\{e\}$, with $\mathbf{Q}$ an extremely simple block diagonal matrix whose diagonal blocks are either of size 1×1 or 2×2. The detailed formulation of $\mathbf{Q}$ will be given in next section.

In addition, when generating (5), apparently, we have an infinite number of choices of the points $\mathbf{r}_{hi}$ and the directions $\hat{h}_i$ for computing the discrete $\mathbf{H}$. However, to connect (5) to (10), we need to keep in mind that the $\mathbf{H}$-points and directions we choose should facilitate *accurate* generation of the $\{e\}$ desired in (5) so that we can march on in time step by step—from $\{e\}$ to $\{h\}$ via (5), and then from $\{h\}$ back to $\{e\}$ through (10).

D. Time Marching

A leap-frog-based time discretization of (5) and (10) clearly yields a time-marching scheme free of matrix solutions as follows:

$$\{h\}^{n+\frac{1}{2}} = \{h\}^{n-\frac{1}{2}} - \mathrm{diag}\left(\left\{\frac{1}{\mu}\right\}\right)\Delta t\mathbf{S}_e\mathbf{Q}\{e\}^n \quad (11)$$

$$\left(\mathrm{diag}\left(\{\epsilon\}\right) + \frac{\Delta t}{2}\mathrm{diag}\left(\{\sigma\}\right)\right)\{e\}^{n+1} = \left(\mathrm{diag}\left(\{\epsilon\}\right) - \frac{\Delta t}{2}\mathrm{diag}\left(\{\sigma\}\right)\right)\{e\}^n + \Delta t\mathbf{S}_h\{h\}^{n+\frac{1}{2}} - \Delta t\{j\}^{n+\frac{1}{2}} \quad (12)$$

where Δt is the time step, and the time instants for $\{e\}$ and $\{h\}$, denoted by superscripts, are staggered by half. Neither (11) nor (12) involves a matrix solution.

Equations (5) and (10) can also be solved in a second-order based way. Taking another time derivative of (10) and substituting (5), we obtain

$$\frac{\partial^2\{e\}}{\partial t^2} + \mathrm{diag}\left(\left\{\frac{\sigma}{\epsilon}\right\}\right)\frac{\partial\{e\}}{\partial t} + \mathbf{S}\{e\} = -\mathrm{diag}\left(\left\{\frac{1}{\epsilon}\right\}\right)\frac{\partial\{j\}}{\partial t} \quad (13)$$

where

$$\mathbf{S} = \mathrm{diag}\left(\left\{\frac{1}{\epsilon}\right\}\right)\mathbf{S}_h\mathrm{diag}\left(\left\{\frac{1}{\mu}\right\}\right)\mathbf{S}_e\mathbf{Q}. \quad (14)$$

It is evident that the above numerical system is also free of matrix solutions with a central-difference based discretization in time. This is because the matrix in front of the second-order time derivative, which is known as mass matrix, and the matrix before the first-order time derivative are both naturally diagonal. Since the matrices are made naturally diagonal in the proposed method, no approximation-based mass-lumping is needed.

It is also worth mentioning that the leap-frog-based time discretization shown in (11) and (12) is the same as the central-difference-based explicit discretization of the second-order system (13). This can be readily seen by writing the counterpart of (12) for evaluating $\{e\}^n$, i.e., replacing n by $n-1$ in (12), subtracting the resultant from (12), and then substituting (11) to replace the $\{h\}^{n+(1/2)} - \{h\}^{n-(1/2)}$ term. Since (11) and (12) are the same as the explicit discretization of (13), we can directly solve (13), which also has only half a number of unknowns. If $\{h\}$ unknowns are needed, they can readily be recovered from $\{e\}$ through (11).

E. Remark

In the framework described above, we expand $\mathbf{E}$ into certain vector basis functions in each element, while sampling the $\mathbf{H}$ unknowns at discrete points to generate desired $\mathbf{E}$ unknowns. One can also switch the roles of the electric and magnetic fields: expand the $\mathbf{H}$ into vector basis functions in each element, while sampling the $\mathbf{E}$ unknowns. Which way to use depends on the convenience for solving a given problem.

III. Proposed Formulations

In this section, we present detailed formulations to realize the aforementioned matrix-free framework with guaranteed accuracy and stability. Since 2-D formulations have been presented in [24], 3-D formulations will be the focus of this section.

A. Accurate Construction of $\mathbf{S}_e$ and $\mathbf{E}$'s Degrees of Freedom

A common choice of the vector basis functions for expanding the fields is the zeroth-order curl-conforming bases (edge elements) [25]. These bases have constant tangential components along the edges where they are defined. The field representation in the traditional FDTD is, in fact, a zeroth-order vector basis representation in an orthogonal cell. However, the zeroth-order vector bases have a constant curl in every element. Using such bases to represent $\mathbf{E}$, the resultant $\mathbf{H}$ is a constant in each element, and the $\mathbf{H}$ is only second-order accurate at the center point of each element. From such discrete $\mathbf{H}$-fields, we cannot

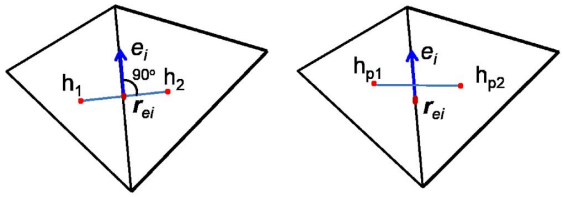


Fig. 1. (a) Locations of $\mathbf{H}$ points required for the accurate evaluation of e at point $\mathbf{r}_e$. (b) Locations of $\mathbf{H}$ points with zeroth-order vector bases.

reversely obtain the $\mathbf{E}$ unknowns associated with the zeroth-order vector bases accurately in an arbitrarily shaped element. To help understand the aforementioned point more clearly, take a 2-D problem discretized into arbitrarily shaped triangular elements as an example. Consider an arbitrary ith edge. With the zeroth-order vector bases to expand $\mathbf{E}$, the e_i shown in (9) has $\hat{e}_i$ the unit vector tangential to the ith edge, and $\mathbf{r}_{ei}$ the center point of the ith edge, as illustrated in Fig. 1. To obtain such an e_i accurately from the discrete $\mathbf{H}$ (now H_z only since the problem is 2-D), the two $\mathbf{H}$-points should be located on the line that is perpendicular to the ith edge and centered at the point $\mathbf{r}_{ei}$, as shown in Fig. 1(a). In this way, the edge is perpendicular to the $\mathbf{H}$-loop (in the plane defined by z-direction and the line normal to the edge), and resides at the center of the loop. As a result, an accurate $\mathbf{E}\cdot\hat{e}_i$ can be obtained from a space derivative of the two $\mathbf{H}$ unknowns. However, using the zeroth-order edge elements, the curl of $\mathbf{E}$ is constant in every element, thus we cannot generate $\mathbf{H}$ at the desired points accurately. From another perspective, we can view the $\mathbf{H}$ obtained at the center point of every element to be accurate. However, in an arbitrary unstructured mesh, the line segment connecting the center points of the two elements sharing an edge may not be perpendicular to the edge, and the two center points may not have the same distance to the edge either, as illustrated in Fig. 1(b).

To overcome the aforementioned problem, we propose to use higher-order curl-conforming vector bases to expand $\mathbf{E}$ in each element. With an order higher than zero, the curl of $\mathbf{E}$ and hence $\mathbf{H}$ is at least a linear function of x, y, and z in each element. With this, the $\mathbf{H}$ can be obtained at an arbitrary point along an arbitrary direction accurately from (5). We hence can use this freedom to choose $\mathbf{H}$ points and directions in such a way that they can reversely generate $\mathbf{E}$ unknowns accurately from (10).

First-order bases are sufficient for use. Certainly, one can employ bases whose order is even higher. This is one of the reasons why the detailed formulations corresponding to the proposed framework are not unique. In this work, first-order bases are used, since they satisfy the need of the proposed matrix-free method and they minimize computational overhead as compared to other bases. For completeness of this paper, in Appendix, we list all the twenty first-order bases in a tetrahedral element [26] together with their degrees of freedom defined in terms of locations $\mathbf{r}_{ei}$ and projection directions $\hat{e}_i$ $(i = 1, 2, \ldots, 20)$.

B. Relationship Between $\{u\}$ and $\{e\}$

The vector $\{u\}$ contains the unknown coefficients of vector basis functions as shown in (3), while vector $\{e\}$ contains the discrete electric fields at $\mathbf{r}_{ei}$ points along $\hat{e}_i$ directions as defined in (9). If $u_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$, then $\{u\} = \{e\}$. Hence, (10) and (5) are directly connected to each other. Among higher-order vector basis functions [26], the vector bases associated with edges satisfy $u_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$ naturally. However, the bases defined on the faces and those inside the element, in general, do not. This problem can be solved by modifying the original higher-order vector bases to make $\{u\} = \{e\}$, as done in [24]. We can also keep the original higher-order vector bases as they are, but find the relationship between $\{u\}$ and $\{e\}$ as follows.

Substituting (3) into (9), we have

$$e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i = \sum u_j (\mathbf{N}_j (\mathbf{r}_{ei}) \cdot \hat{e}_i) \tag{15}$$

from which we obtain

$$\{e\} = \mathbf{P}\{u\} \tag{16}$$

where $\mathbf{P}$ matrix obviously has the following entries:

$$\mathbf{P}_{ij} = \mathbf{N}_j(\mathbf{r}_{ei}) \cdot \hat{e}_i. \tag{17}$$

The $\mathbf{P}$ is of size N_e but an extremely simple matrix—It is a block diagonal matrix with each diagonal block of size either 1 or 2. To be specific, for the vector basis function i whose degree of freedom is associated with edges, the $\mathbf{P}_{ii} = 1$ and elsewhere in the ith row $\mathbf{P}_{ij} = 0$; for the vector basis function i whose degree of freedom is *not* associated with edges, it is either defined on faces or inside the element. Such a basis function comes in as a pair, for which there are two nonzero elements on the ith row of $\mathbf{P}$, and two nonzero elements on the $(i+1)$th row of $\mathbf{P}$, forming a 2×2 diagonal block in $\mathbf{P}$ as the following

$$\mathbf{P}_i = \begin{bmatrix} 1 & \mathbf{N}_{i+1}(\mathbf{r}_{ei}) \cdot \hat{e}_i \\ \mathbf{N}_i(\mathbf{r}_{e,i+1}) \cdot \hat{e}_{i+1} & 1 \end{bmatrix}. \tag{18}$$

The off-diagonal terms in the above do not vanish because for face or internal degrees of freedom, the basis function pair associated with each point are not perpendicular to each other in terms of the vector basis's direction. Overall, the $\mathbf{P}$ can be written as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 & 0 & \ldots & 0 \\ 0 & \mathbf{P}_2 & 0 & \ldots & 0 \\ 0 & 0 & \mathbf{P}_3 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & \mathbf{P}_{\ldots} \end{bmatrix} \tag{19}$$

where each diagonal block $\mathbf{P}_i$ is equal to either 1 or a 2×2 matrix shown in (18), which can be readily inverted to obtain $\mathbf{P}^{-1}$, denoted by $\mathbf{Q}$. Obviously, $\mathbf{Q}$ is also a block diagonal matrix whose diagonal blocks are of size either 1 or 2. As a result, we find a closed-form relationship between $\{u\}$ from $\{e\}$ as

$$\{u\} = \mathbf{Q}\{e\}. \tag{20}$$

Equation (5) hence can be rewritten as

$$\mathbf{S}_e\mathbf{Q}\{e\} = -\text{diag}\left(\{\mu\}\right) \frac{\partial\{h\}}{\partial t}. \tag{21}$$

Thus, (10) and (5) are connected to each other.

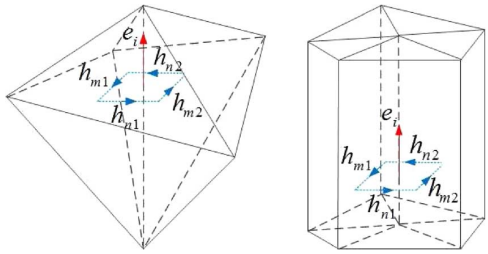


Fig. 2. **H** points and directions for generating e_i.

C. Accurate Construction of $\mathbf{S}_h$ and Choice of **H**'s Points and Directions

To construct (10) accurately, we propose to use an **H**-loop uniquely defined for each **E**'s degree of freedom to obtain the **E** desired in (5). This loop centers each **E**'s degree of freedom, and is also positioned perpendicular to the **E**'s degree of freedom. This **H**-loop can be chosen in its simplest manner: a 1-D line segment in 2-D settings, and a 2-D rectangular loop centering and normal to the **E**'s degree of freedom in 3-D problems, as shown in Fig. 2. Regardless of the shape of the element, such a rectangular loop can always be defined for each **E** unknown. Along this loop, we select the middle points of the four sides as **H**-points and the four unit vectors tangential to each side as **H**-directions to generate $\{h\}$. As a result, each **E** unknown e_i is associated with four **H**-points and directions. These **H**-points are all located *inside* the elements that share the **E** unknown, instead of being selected on the faces of the elements. In this way, each **H** point is located only in one element, and hence the **H**-field at the point can be readily found from (5). The set of **H**-points and **H**-directions defined for each e_i makes the whole set of **H**-points denoted by $\{\mathbf{r}_{hi}\}$, and the whole set of **H**-directions denoted by $\{\hat{h}_i\}$ $(i = 1, 2, \ldots, N_h)$.

With the aforementioned choice of **H**-points and directions, the $\hat{e}_i \cdot \nabla \times \mathbf{H}$ in (8) can be accurately discretized with second-order accuracy as the following

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = (h_{m1} + h_{m2})/l_{im} + (h_{n1} + h_{n2})/l_{in} \quad (22)$$

where l_{im} is the distance between h_{m1} and h_{m2}, while l_{in} is the distance between h_{n1} and h_{n2} as illustrated in Fig. 2. With (22), we obtain

$$\mathbf{S}_{h,ij} = \frac{1}{l_{ij}} \quad (23)$$

where j denotes the global index of the **H**-point associated with the e_i, and l_{ij} is simply two times the distance between the **H**-point $(\mathbf{r}_{hj})$ and the **E**-point $(\mathbf{r}_{ei})$. Each row of $\mathbf{S}_h$ has only four nonzero elements.

Obviously, there is no need to construct a dual mesh for **H** as the **H**-points and **H**-directions we select are individually defined for each **E** unknown, which do not make a mesh. In addition, regardless of the choice of **H**-points and directions, there is no difficulty in generating corresponding $\{h\}$ from (5) *accurately*, due to the use of higher-order basis functions.

D. Imposing Boundary Conditions

The proposed method, in its first-order form (11)–(12), conforms to that of the FDTD numerical system; in its second-order form (13), conforms to the second-order wave equation based TDFEM. Hence, the boundary conditions in the proposed method can be implemented in the same way as those in the TDFEM and FDTD. Below we provide more details.

For closed-region problems, the perfect electric conductor (PEC), the perfect magnetic conductor (PMC), or other nonzero prescribed tangential **E** or tangential **H** are commonly used at the boundary. To impose prescribed tangential **E** at N_b boundary points, in (5), we simply set the $\{e\}$ entries at the N_b points to be the prescribed value, and keep the size of $\mathbf{S}_e$ the same as before to produce all N_h discrete **H** from the N_e discrete **E**. In (10), since the $\{e\}$ entries at the N_b points are known, the updating of (10) only needs to be performed for the rest $(N_e - N_b)$ $\{e\}$ entries. As a result, we can remove the N_b rows from $\mathbf{S}_h$ corresponding to the N_b boundary **E** fields, while keeping the column dimension of $\mathbf{S}_h$ the same as before. The above treatment, from the perspective of the second-order system shown in (13), is the same as keeping just $(N_e - N_b)$ rows of $\mathbf{S}$, providing the full-length $\{e\}$ (with the boundary entries specified) for the $\{e\}$ multiplied by $\mathbf{S}$, but taking only the $N_e - N_b$ rows of all the other terms involved in (13). To impose a PMC boundary condition, the total **E** unknown number is N_e without any reduction. Equation (5) is formulated as it is since the **H**-points having the PMC boundary condition can be placed outside the computational domain. As for (10), there is no need to make any change either since the tangential **H** is set to be zero outside the computational domain. The end result is the same as a TDFEM numerical system subject to the second-kind boundary condition.

For open-region problems, the framework of (5) and (10) in the proposed method is conformal to that of the FDTD. As a result, the various absorbing boundary conditions that have been implemented in FDTD such as the commonly used PML (perfectly matched layer) can be implemented in the same way in the proposed matrix-free method.

IV. Time Marching Free of Matrix-Solution With Guaranteed Stability

A leap-frog-based time marching shown in (11)–(12) as well as a central-difference based time discretization of (13) is absolutely matrix-free, i.e., free of a matrix solution. However, both are absolutely unstable since the curl-curl operator here is an unsymmetrical matrix. This is not only true for the proposed method but also true for any method whose curl operation on one field unknown is not the reciprocal of the curl operation on the other field unknown. To prove, we can perform a stability analysis of (11)–(12) and (13) [27], [28]. The z-transform of the central-difference based time marching of (13), or (11)–(12) after eliminating $\{h\}$, results in the following equation:

$$(z-1)^2 + \Delta t^2 \lambda z = 0 \quad (24)$$

where λ is the eigenvalue of $\mathbf{S}$. The two roots of (24) can be readily found as

$$z_{1,2} = \frac{2 - \Delta t^2 \lambda \pm \sqrt{\Delta t^2 \lambda (\Delta t^2 \lambda - 4)}}{2}. \quad (25)$$

If $\mathbf{S}$ is Hermitian positive semidefinite like that resulting from TDFEM or FDTD in an orthogonal grid, all its eigenvalues are nonnegative real. Thus, we can always find a time step to make

z in (25) bounded by 1, and hence the explicit simulation of (13) as well as (11)–(12) is stable. Such a time step satisfies $\Delta t \leq 2/\sqrt{\lambda_{\max}}$, where $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{S}$, which is also $\mathbf{S}$'s spectral radius. However, if $\mathbf{S}$ is not symmetrical, which is the case in the proposed method and many existing nonorthogonal FDTD methods, its eigenvalues either are real (can be negative) or come in complex-conjugate pairs. For complex-valued eigenvalues λ as well as negative ones, the two roots z_1 and z_2 shown in (25) 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 the explicit time-domain simulation of (13) and (12) must be unstable.

However, if we choose $\mathbf{S}_h = \mathbf{S}_e^T$ to make $\mathbf{S}$ symmetric, the accuracy cannot be guaranteed in a general unstructured mesh. This dilemma is solved as follows without sacrificing the matrix-free merit of the proposed method. Basically, we can start with the following backward-difference based discretization of (13) [17]:

$$\begin{aligned}&\{e\}^{n+1} - 2\{e\}^n + \{e\}^{n-1} \\ &\quad + \triangle t \mathrm{diag}\left(\left\{\frac{\sigma}{\epsilon}\right\}\right)\left(\{e\}^{n+1} - \{e\}^n\right) + \triangle t^2 \mathbf{S}\{e\}^{n+1} \\ &\quad = -\triangle t^2 \mathrm{diag}\left(\left\{\frac{1}{\epsilon}\right\}\right)\left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}\end{aligned} \tag{26}$$

where the $\{e\}$ associated with $\mathbf{S}$ is chosen at the $(n+1)$th time step instead of the nth step. Performing a stability analysis of (26) for lossless cases, we find the two roots of z as

$$z_{1,2} = \frac{1}{1 \pm j\Delta t\sqrt{\lambda}}. \tag{27}$$

As a result, the z can still be bounded by 1 even for an infinitely large time step. However, this does not mean the backward difference is unconditionally stable since now the λ can be complex-valued or even negative. To make the magnitude of (27) bounded by 1, we find that the time step needs to satisfy the following condition

$$\triangle t > 2\frac{\left|\mathrm{Im}(\sqrt{\lambda})\right|}{\left|\sqrt{\lambda}\right|^2} \tag{28}$$

where $\mathrm{Im}(\cdot)$ denotes the imaginary part of $(\cdot)$. It is obvious to see that the scheme is stable for large time step, but not stable for small time step. Such a requirement happens to align with preferred choices of time step, since a large time step is desired for an efficient simulation.

Rearranging the terms in (26), we obtain

$$\begin{aligned}\tilde{\mathbf{D}}\{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} + \triangle t \mathrm{diag}\left(\left\{\frac{\sigma}{\epsilon}\right\}\right)\{e\}^n \\ -\triangle t^2 \mathrm{diag}\left(\left\{\frac{1}{\epsilon}\right\}\right)\left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}\end{aligned} \tag{29}$$

where

$$\tilde{\mathbf{D}} = \mathbf{I} + \triangle t \mathrm{diag}\left(\left\{\frac{\sigma}{\epsilon}\right\}\right) + \triangle t^2 \mathbf{S}. \tag{30}$$

Since $\tilde{\mathbf{D}}$ is not diagonal, (29) requires a matrix solution. To avoid that, we can solve this problem as follows.

Let the diagonal part of $\tilde{\mathbf{D}}$ be $\mathbf{D}$, which means

$$\mathbf{D} = \mathbf{I} + \triangle t \mathrm{diag}\left(\left\{\frac{\sigma}{\epsilon}\right\}\right). \tag{31}$$

Front multiplying both sides of (29) by $\mathbf{D}^{-1}$, we obtain

$$(\mathbf{I} + \tilde{\mathbf{M}})\{e\}^{n+1} = \mathbf{D}^{-1}\{f\} \tag{32}$$

where $\{f\}$ is the right hand side of (29), and

$$\tilde{\mathbf{M}} = \Delta t^2 \mathbf{D}^{-1}\mathbf{S}. \tag{33}$$

Although (29) permits the use of any large time step, when we choose the time step based on that of a conventional explicit method, the time step satisfies

$$\Delta t^2 < \frac{1}{\|\mathbf{S}\|} \tag{34}$$

and therefore

$$\Delta t^2 \|\mathbf{S}\| < 1. \tag{35}$$

This time step is also the time step required by accuracy when space step is determined by accuracy. Since $\mathbf{D}$ in (31) is diagonal, the norm of its inverse can be analytically evaluated as

$$\|\mathbf{D}^{-1}\| = \frac{1}{\min_{1\leq i\leq N_e}(1 + \Delta t\sigma_i/\epsilon_i)} = 1. \tag{36}$$

We therefore obtain from (35) and (36)

$$\|\tilde{\mathbf{M}}\| = \Delta t^2 \|\mathbf{D}^{-1}\mathbf{S}\| \leq \Delta t^2 \|\mathbf{D}^{-1}\|\|\mathbf{S}\| < 1. \tag{37}$$

As a result, the inverse of $\mathbf{I} + \tilde{\mathbf{M}}$ can be explicitly represented as a series expansion

$$(\mathbf{I} + \tilde{\mathbf{M}})^{-1} = \mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \tilde{\mathbf{M}}^3 + \ldots \tag{38}$$

which can be truncated after the first few terms without sacrificing accuracy due to (37). Thus, the system matrix has an explicit inverse, and hence no matrix solution is required in the proposed method. The final update equation becomes

$$\{e\}^{n+1} = \left(\mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \ldots + (-\tilde{\mathbf{M}})^k\right)\mathbf{D}_i\{f\} \tag{39}$$

where $\mathbf{D}_i$ is a diagonal matrix which is $\mathbf{D}$'s inverse. The number of terms k is guaranteed to be small (less than 10) since (37) holds true, and the central-difference-based time step (34) is usually not chosen right at the boundary, $1/\|\mathbf{S}\|$, but smaller for better sampling accuracy. Notice that the spectral radius of $\tilde{\mathbf{M}}$, as revealed in (37), is essentially the square of the ratio of the actual time step used (Δt) to the largest time step permitted by the stability of a conventional explicit scheme $(\sim 1/\|\mathbf{S}\|)$. It is a constant irrespective of the mesh quality. Therefore, the convergence of (38) is guaranteed and the convergence rate does not depend on the mesh quality. Notice that using (38) does not change the stability analysis since it is used to obtain the inverse of system matrix, which does not change the backward difference based time marching scheme.

The computational cost of (39) is k sparse matrix-vector multiplications since each term can be computed from the previous term. For example, if we first compute $y = \mathbf{D}_i\{f\}$, then the second term in (39) can be obtained from $-\tilde{\mathbf{M}}y$. Let the resultant be y. The third term relating to $\tilde{\mathbf{M}}^2$ is nothing but $\tilde{\mathbf{M}}y$. Therefore, the cost for computing each term in (39) is the cost of multiplying $\tilde{\mathbf{M}}$ by the vector obtained at the previous step, thus efficient.

V. Relationship with FDTD

In a regular orthogonal grid and with the zeroth-order vector bases, the proposed method reduces exactly to the FDTD. This is very different from the mixed $\mathbf{E}-\mathbf{B}$ formulation like [29] where mass lumping has to be used to prove equivalency. To explain, for a 2-D rectangular grid and a 3-D brick-element based discretization, with a zeroth-order edge vector basis used in each rectangular or brick element, the operation of $\mathbf{S}_e\{e\}$ in the proposed method is the same as how the curl of $\mathbf{E}$ is discretized in the FDTD; and the operation of $\mathbf{S}_h\{h\}$ with $\mathbf{S}_h = \mathbf{S}_e^T$ is the same as how the curl of $\mathbf{H}$ is discretized in the FDTD. Furthermore, since $\mathbf{S}_h = \mathbf{S}_e^T$ naturally satisfies in an orthogonal grid, the resulting numerical system is symmetric and positive semidefinite. Hence the original leap-frog explicit time marching is stable without any need for special treatment. That is also why in a traditional FDTD with an orthogonal grid, an explicit time marching is never observed to be absolutely unstable because the system matrix is symmetric.

To see the above point more clearly, take the 2-D rectangular grid as an example. The $\{e\}$ is simply a union of $\mathbf{E}\cdot\hat{e}_i$ at the center point of each edge, with $\hat{e}_i$ being either x or y along each edge; and the $\{h\}$ is nothing but the vector containing H_z at the center point of each rectangular patch. Each row of $\mathbf{S}_e$ has four nonzero elements as each element has four bases. Multiplying the ith row of $\mathbf{S}_e$ by $\{e\}$ is nothing but

$$\frac{e_m - e_n}{W} - \frac{e_p - e_q}{L} \tag{40}$$

where m, n, p, q are the global indexes of the four edge basis functions in the rectangular element where the $\mathbf{H}$ point is located, and W and L are the two side lengths of the rectangular element. It is evident that (40) is the same as that performed in the FDTD to produce the H_z at the center of each $\mathbf{E}$-loop. With $\mathbf{S}_h = \mathbf{S}_e^T$, the operation of $\mathbf{S}_h\{h\}$ is to do

$$\frac{1}{l_j}h_{p1} - \frac{1}{l_j}h_{p2} = \frac{h_{p1} - h_{p2}}{l_j} \tag{41}$$

where l_j is simply the length of the side that is perpendicular to edge j in a rectangular element. Obviously, the above is the same as that used in the FDTD to calculate $\mathbf{E}$ fields, which is an accurate discretization of $\nabla\times\mathbf{H}$ of second-order accuracy at the center point of an edge for $\mathbf{E}$ along the edge.

In addition, even in an orthogonal grid, the implementation of the proposed method is more convenient, since no dual grid is needed. After $\mathbf{S}_e$ is formed for the grid, $\mathbf{S}_h$ is known as $\mathbf{S}_e^T$ without any construction cost. For unstructured meshes, the FDTD method would fail, whereas the proposed method is accurate and stable regardless of how irregular and unstructured the mesh is.

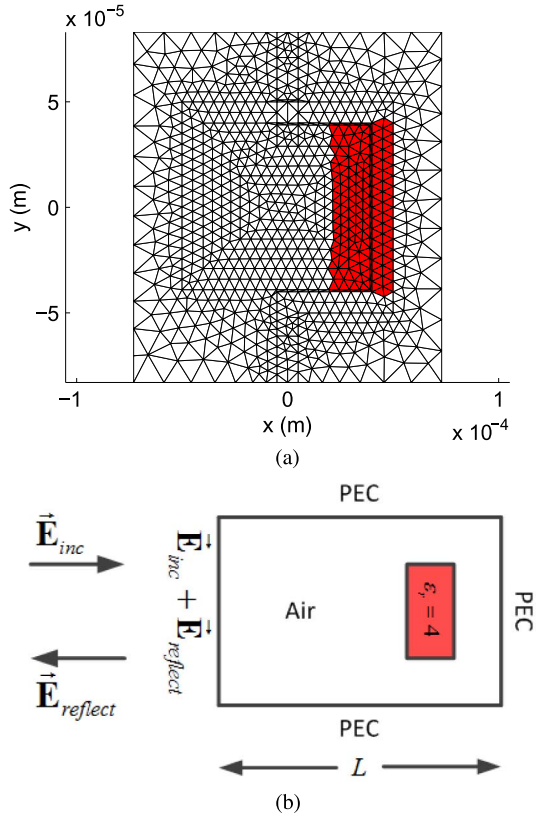


Fig. 3. Simulation of wave propagation and reflection in a 2-D triangular mesh. (a) Mesh. (b) Illustration of incident wave and truncation boundary conditions.

VI. Numerical Results

In this section, we simulate a variety of 2- and 3-D unstructured meshes to demonstrate the validity and generality of the proposed matrix-free method in analyzing arbitrarily shaped structures and materials discretized into unstructured meshes. The accuracy of the proposed method is validated by comparing with both analytical solutions and the TDFEM method that is capable of handling unstructured meshes but not matrix-free.

A. 2-D Triangular Mesh

The first example is a wave propagation and reflection problem in an 2-D triangular mesh shown in Fig. 3(a). Some mesh elements are very skewed due to fine features in a narrow gap whose size is less than a few μm. The dielectric constant is $\epsilon_r = 4$ in the red shaded region and 1 elsewhere. The incident $\mathbf{E}$ is specified as $\hat{y}f(t - x/c)$, where $f(t) = 2(t-t_0)\exp(-(t-t_0)^2/\tau^2)$, $\tau = 8.0\times10^{-13}$ s, $t_0 = 4\tau$ s, and c denotes the speed of light. The top, bottom and right boundaries are terminated by PEC, while the left boundary is truncated by the sum of the incident and reflected $\mathbf{E}$ fields as illustrated in Fig. 3(b). Since the left boundary is not close to the dielectric discontinuity, the reflected field at the left boundary can be analytically approximated as $-\hat{y}f(t - x_0/c - 2L/c)$, where x_0 is the x-coordinate at the left boundary, and L is the width of the computational domain.

In the proposed method, the number of expansion terms used is 9 in (38). For comparison, we simulate the same example by TDFEM since it is capable of handling unstructured meshes. The time step used in both methods is 5×10^{-16} s. In Fig. 4(a), the electric fields at two

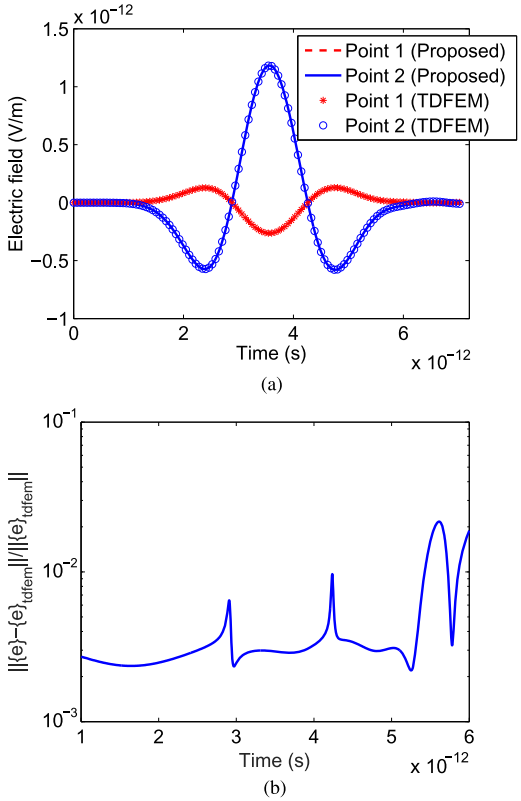


Fig. 4. Simulation of a 2-D triangular mesh. (a) Electric fields at two points. (b) *Entire* solution error versus time.

points $\mathbf{r}_{p1} = (-5.912 \times 10^{-5}, -7.131 \times 10^{-5}, 0)$ and $\mathbf{r}_{p2} = (-6.325 \times 10^{-5}, -6.434 \times 10^{-5}, 0)$ randomly selected are plotted in comparison with TDFEM results. The directions of the two fields are respectively $\hat{e}_{p1} = 0.979\hat{x} - 0.206\hat{y}$, and $\hat{e}_{p2} = 0.463\hat{x} - 0.886\hat{y}$. Excellent agreement can be observed with TDFEM results. Such an agreement is also observed at all points for all time. As shown in Fig. 4(b), the entire solution error as compared with the TDFEM solution is less than 2% at all time instants. A few peak errors are due to the comparison with close-to-zero fields. The entire solution error is defined by

$$\text{Error}_{\text{entire}}(t) = \frac{\|\{e\}_{\text{this}}(t) - \{e\}_{\text{ref}}(t)\|}{\|\{e\}_{\text{ref}}(t)\|} \qquad (42)$$

where $\{e\}_{\text{this}}(t)$ denotes the entire unknown vector $\{e\}$ of length N_e solved from the proposed method, and $\{e\}_{\text{ref}}(t)$ denotes the reference solution, which is TDFEM result in this example.

B. Wave Propagation in a 3-D Box Discretized into Tetrahedral Mesh

A 3-D box discretized into tetrahedral elements is simulated in free space. The mesh details are shown in Fig. 5. The discretization results in 544 edges and 350 elements. To investigate the accuracy of the proposed method in such a mesh, we consider that the most convincing comparison is a comparison with analytical solution. We hence study a free-space wave propagation problem whose analytical solution is known. To simulate such an open-region problem, we impose an analytical boundary condition, i.e., the known value of tangential $\mathbf{E}$, on the outermost boundary of the problem; we then numerically simulate

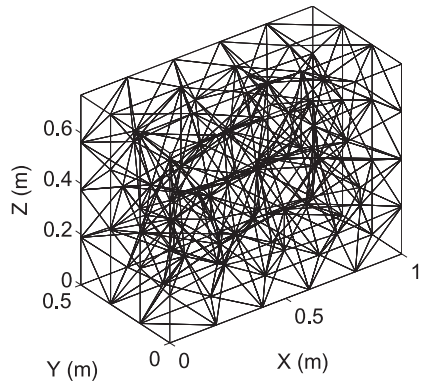


Fig. 5. Illustration of the tetrahedron mesh of a 3-D structure.

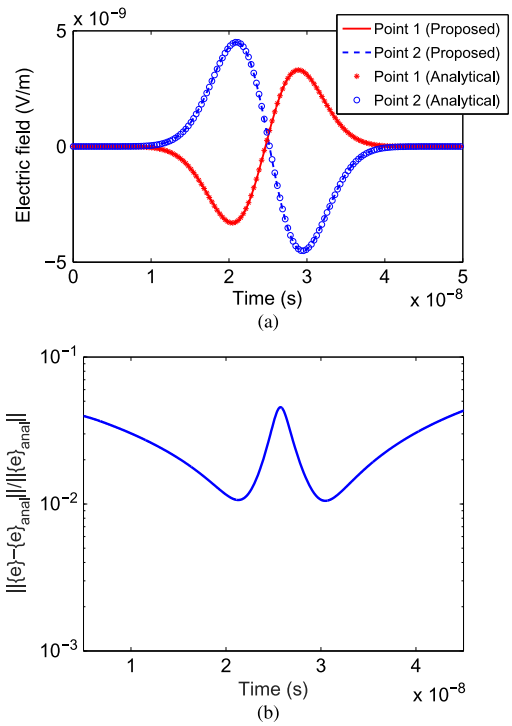


Fig. 6. Simulation of a 3-D box discretized into tetrahedral elements. (a) Simulated two electric fields in comparison with analytical results. (b) Entire solution error for all $\mathbf{E}$ unknowns versus time.

the fields inside the computational domain and correlate results with the analytical solution.

The structure is illuminated by a plane wave having $\mathbf{E} = \hat{y}f(t - x/c)$, where $f(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, $\tau = 6.0 \times 10^{-9}$ s, and $t_0 = 4\tau$. The time step used in the proposed method is $\Delta t = 1.6 \times 10^{-11}$ s, which is the same as what a traditional central-difference based TDFEM has to use for stability. The number of expansion terms is 9 in (38). In Fig. 6(a), we plot the first and 1832th entry randomly selected from the unknown $\{e\}$ vector, which represent $\mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$, with $i = 1$, and 1832 respectively. From Fig. 6(a), it can be seen clearly that the electric fields solved from the proposed method have an excellent agreement with analytical results. To further verify the accuracy of the proposed method in the entire computational domain, we assess the entire solution error (42) as a function of time, where the reference solution is analytical result $\{e\}_{\text{anal}}(t)$. In Fig. 6(b), we plot $\text{Error}_{\text{entire}}(t)$ across the

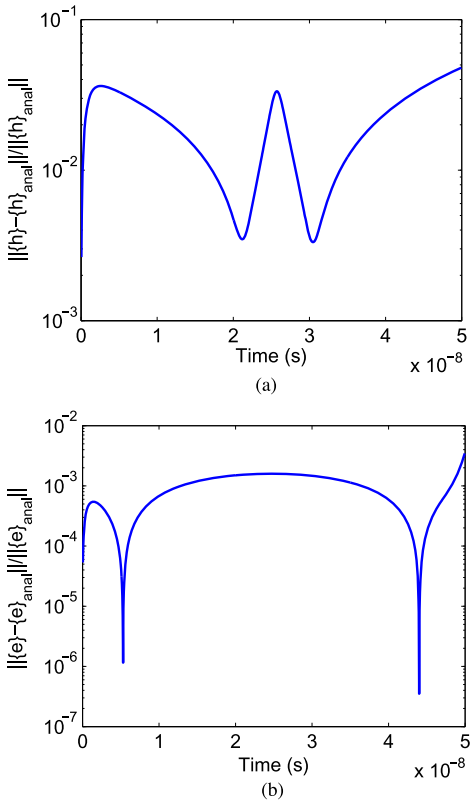


Fig. 7. (a) Entire solution error versus time of all $\mathbf{H}$ unknowns obtained from $\mathbf{S}_e$-rows of equations. (b) Entire solution error versus time of all $\mathbf{E}$ obtained from $\mathbf{S}_h$-rows of equations.

whole time window in which the fields are not zero. It is evident that less than 4% error is observed at each time instant, demonstrating the accuracy of the proposed method. The center peak in Fig. 6(b) is due to a comparison with close to zero fields.

In addition to the accuracy of the entire method, we have also examined the accuracy of the individual $\mathbf{S}_e$, and $\mathbf{S}_h$ separately, since each is important to ensure the accuracy of the whole scheme. First, to solely assess the accuracy of $\mathbf{S}_e$, we perform the time marching of (5) only without (10) by providing an analytical $\{e\}$ to (5) at each time step. The resultant $\{h\}$ is then compared to analytical $\{h\}_{\text{anal}}$ at each time step. As can be seen from Fig. 7(a) where the following $\mathbf{H}$-error

$$\frac{\|h(t) - h_{\text{anal}}(t)\|}{\|h_{\text{anal}}(t)\|} \tag{43}$$

is plotted with respect to time, the error of all $\mathbf{H}$ unknowns is less than 3% across the whole time window, verifying the accuracy of $\mathbf{S}_e$.

Similarly, in order to examine the accuracy of $\mathbf{S}_h$, we perform the time marching of (10) only without (5) by providing an analytical $\{h\}$ to (10) at each time step. The relative error of all $\mathbf{E}$ unknowns shown in (42) as compared to analytical solutions is plotted with time in Fig. 7(b). Again, very good accuracy is observed across the whole time window, verifying the accuracy of $\mathbf{S}_h$.

C. Wave Propagation in a Sphere Discretized into Tetrahedral Mesh

The third example is a sphere discretized into tetrahedral elements in free space, whose 3-D mesh is shown in Fig. 8.

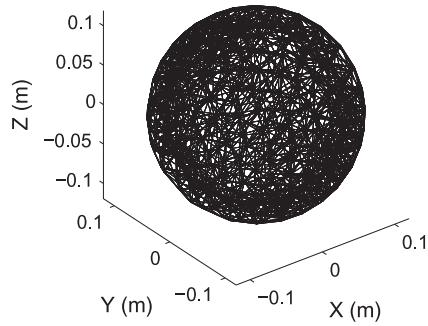


Fig. 8. Illustration of the tetrahedron mesh of a sphere structure.

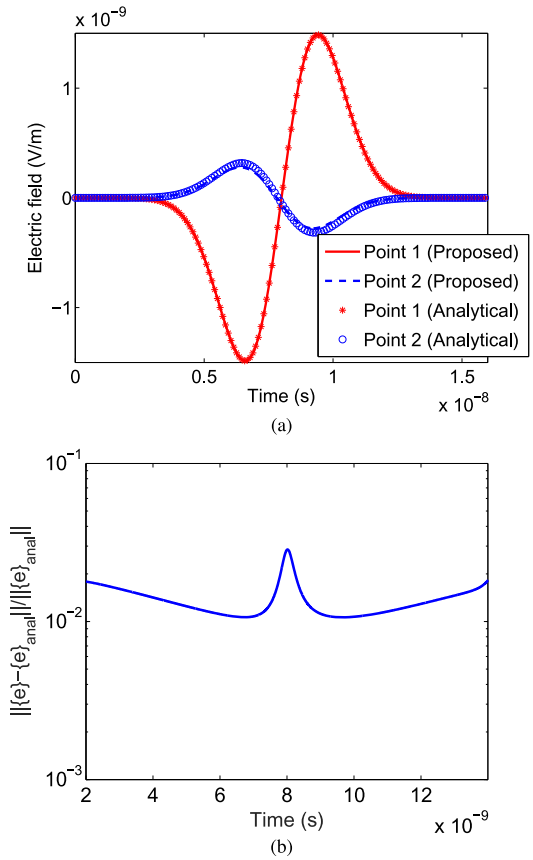


Fig. 9. Simulation of a 3-D sphere discretized into tetrahedral elements. (a) Two electric fields in comparison with analytical results. (b) Entire solution error for all $\mathbf{E}$ unknowns versus time.

The discretization results in 3183 edges and 1987 tetrahedrons. Again, we set up a free-space wave propagation problem in the given mesh to validate the accuracy of the proposed method against analytical results. The incident $\mathbf{E}$ has the same form as that of the first example, but with $\tau = 2.0 \times 10^{-9}$ s in accordance with the new structure's dimension. The outermost boundary of the mesh is truncated by analytical $\mathbf{E}$ fields. The time step used is $\Delta t = 2.0 \times 10^{-12}$ s, which is the same as that used in a traditional TDFEM method. The number of expansion terms is 9 in (38). The two degrees of freedom of the electric field, whose indices in vector $\{e\}$ are 1 and 9762, respectively, are plotted in Fig. 9(a) in comparison with analytical data. Excellent agreement can be observed. In Fig. 9(b), we plot the entire solution error shown in (42) versus time. Less than 3% error is observed in the entire time window. It is evident that the proposed method is not just accurate at certain points, but

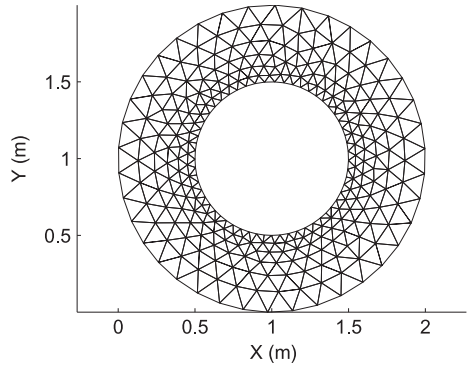


Fig. 10. Top view of the triangular prism mesh of an coaxial cylinder structure.

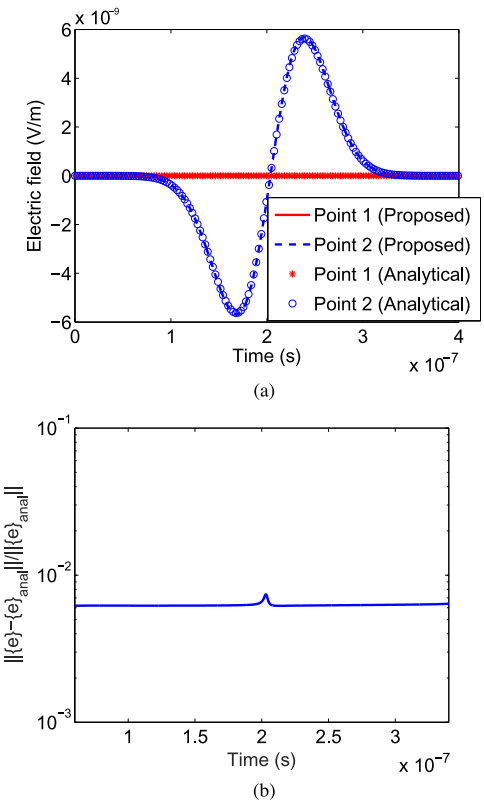


Fig. 11. Simulation of a 3-D coaxial cylinder discretized into triangular prism elements. (a) Two electric fields in comparison with analytical results. (b) Entire solution error for all $\mathbf{E}$ unknowns versus time.

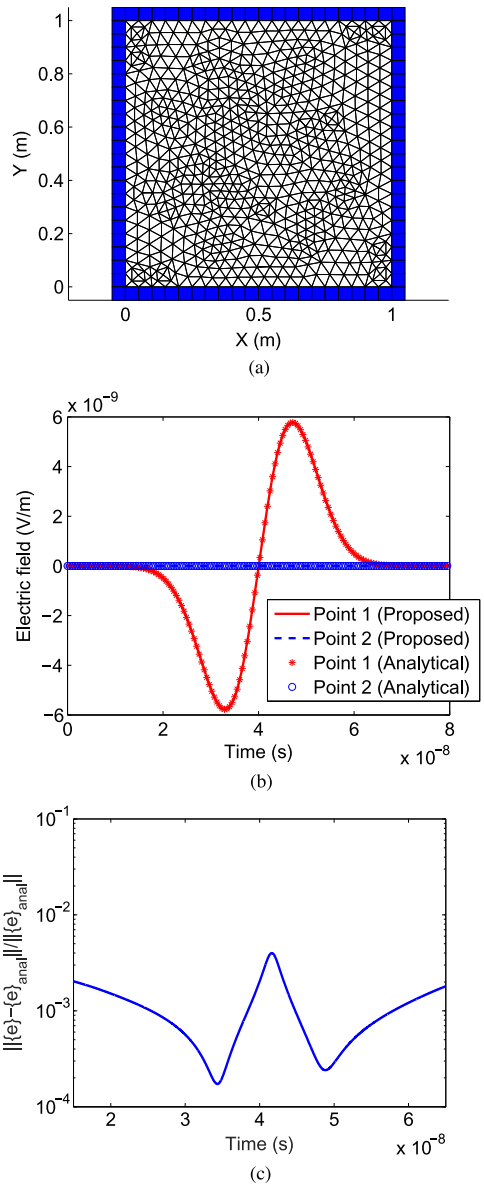


Fig. 12. Simulation of a mesh having different types of elements. (a) Illustration of the mesh. (b) Two electric fields in comparison with analytical results. (c) Entire solution error for all $\mathbf{E}$ unknowns versus time.

accurate at all points in the computational domain for all time instants simulated.

D. Coaxial Cylinder Discretized Into Triangular Prism Mesh

The fourth example has an irregular triangular prism mesh, the top view of which is shown in Fig. 10. The structure has two layers of triangular prism elements (into the paper) with each layer being 0.05 m thick. The discretization results in 3092 edges and 1038 triangular prisms. Both the innermost and outermost boundaries are terminated by exact absorbing boundary condition, which is the analytical tangential $\mathbf{E}$ on the boundary. The incident $\mathbf{E}$ has the same form as that in the first example, but with $\tau = 5.0 \times 10^{-8}$ s. The Δt used is 2.0×10^{-11} s and the number of expansion terms is 9. Two observation points, whose indices in vector $\{e\}$ are 1 and 11 272 respectively, are chosen to plot the electric fields in Fig. 11(a). Excellent agreement with analytical solutions can be observed. In Fig. 11(b), we plot the entire solution error shown in (42) versus time in comparison with the reference results which are analytical solutions. Again, excellent accuracy (less than 0.7% error) is observed at all points in the computational domain for all time instants simulated.

E. Mesh With Mixed Elements

We have examined the capability of the proposed method in handling meshes made of different types of elements. This mesh is illustrated in Fig. 12(a), which consists of 1312 triangular elements in the center and 84 rectangular elements surrounding it. In each triangular element, there are eight first-order vector bases; and in each rectangular element, there are 12 first-order vector bases. The interface between a rectangular and a triangular element is an edge, where the degrees of freedom from both elements are shared in common to ensure the tangential

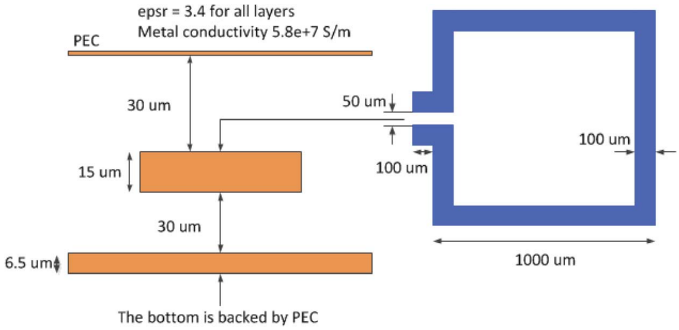


Fig. 13. Illustration of materials and geometry of a package inductor.

continuity of the fields. A wave propagation problem is simulated in this mixed-element mesh. The incident field is a plane wave having $\mathbf{E} = \hat{y}2(t - t_0 - x/c)\exp(-(t - t_0 - x/c)^2/\tau^2)$, where $\tau = 10^{-8}$ s, and $t_0 = 4\tau$. The time step used is $\Delta t = 10^{-11}$ s. In Fig. 12(b), the electric fields at two randomly selected points are plotted in comparison with analytical data. Excellent agreement can be observed. In Fig. 12(c), the entire solution error is plotted as a function of time. Again, excellent accuracy is observed, which verifies the capability of the proposed method in handling meshes having mixed types of elements. Such a capability also facilities a convenient implementation of various absorbing boundary conditions such as the perfectly matched layer.

F. S-Parameter Extraction of a Lossy Package Inductor

In this example, we simulate a package inductor made of lossy conductors of conductivity 5.8e+7 S/m, and embedded in a dielectric material of relative permittivity 3.4. Its geometry and material parameters are illustrated in Fig. 13. The inductor is discretized into five layers of triangular prism elements, the thickness of each of which is 6.5, 30, 6.5, 8.5, and 30 μm from bottom to top, respectively. The top view of the mesh is shown in Fig. 14(a). The boundary conditions are PEC on the top and at the bottom, and PMC on the other four sides. A current source is launched respectively at the two ports of the inductor. It has a Gaussian derivative pulse of $2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, with $\tau = 0.5 \times 10^{-10}$ s, and $t_0 = 4\tau$. The number of expansion terms is 10 used in this simulation. The voltages obtained at both ports with port 1 (upper port) excited and port 2 open are plotted in Fig. 14(b) in comparison with the TDFEM results. Excellent agreement can be observed. The S-parameters are also extracted and compared with those generated from the TDFEM. Very good agreement can be seen from Fig. 14(c) and (d) across the entire frequency band.

G. CPU Time and Memory Comparison

Among existing time-domain methods for handling unstructured meshes, the TDFEM only requires a single mesh like the proposed method. The TDFEM also has guaranteed stability and accuracy, and it ensures the tangential continuity of the fields across material interfaces. We hence choose the TDFEM to benchmark the performance of the proposed method.

The example considered is a large-scale example having millions of unknowns, since small examples are not challenging to solve, which is true to almost every time-domain method. The computational domain is a circular cylinder of radius 1 m and height 5 m, which is discretized into 25 layers of triangular prism elements. The thickness of each layer is 0.02 m. The incident field is a plane wave having $\mathbf{E} = \hat{y}2(t - t_0 - x/c)\exp(-(t-t_0-x/c)^2/\tau^2)$, where $\tau = 10^{-8}$ s, and $t_0 = 4\tau$. The time step used is $\Delta t = 8 \times 10^{-12}$ s, which is the same in the TDFEM and the proposed method. The number of expansion terms used in the proposed method is nine in (38).

The zeroth-order vector bases are employed in the TDFEM, whereas the first-order bases are used in the proposed method. This comparison is, in fact, disadvantageous to the proposed method since the sparse pattern resulting from a higher-order-bases based discretization is much more complicated and the system matrix has many more nonzeros, as compared to the zeroth-order-based discretization. However, if the proposed method is able to show advantages even for such a disadvantageous comparison, then its efficiency gain over the same-order TDFEM would become even more obvious.

The triangular prism discretization results in 3 718 990 $\mathbf{E}$ unknowns in the zeroth-order TDFEM. We find that the TDFEM simulation cannot be performed on our desktop PC that has 16-GB memory due to the TDFEM's large memory requirement. This is because although the explicit TDFEM only requires solving a mass matrix, which is sparse and simple, its $\mathbf{L}$ and $\mathbf{U}$ factors are generally dense. Although the mass matrix is time independent, and hence we only need to factorize it once. The TDFEM still has to be equipped with sufficient memory to store $\mathbf{L}$ and $\mathbf{U}$ factors to carry out the following backward and forward substitutions for the matrix solution at each time step. Certainly, iterative solvers can be used to reduce memory usage, however, they are not cost-effective in time-domain analysis since many right hand sides need to be simulated, and the number of right hand sides is equal to the number of time steps.

We hence find a computer that has 128-GB memory so that the TDFEM simulation can be successfully performed on this example. On this computer, it takes the TDFEM 2109.44 s and more than 72-GB memory to finish the LU factorization of the mass matrix. The CPU time cost at each time marching step is 9.31 s, which is one backward and forward substitution time. For a fair comparison, a similar number of unknowns is generated in the proposed method. The resulting system matrix size is 3 741 700. In contrast to the 2109.44 s cost by TDFEM for factorization, the proposed method has *no* factorization cost since it is free of matrix solution. In contrast to the 72-GB memory required by the TDFEM, the proposed method only takes 6.2-GB memory to store the sparse matrices, as it does not need to store $\mathbf{L}$ and $\mathbf{U}$ since the mass matrix is diagonal. The CPU run time of the proposed method at each time step is 3.76 s, which is spent on a few matrix-vector multiplications. From the aforementioned comparison, the computational efficiency of the proposed method can be clearly seen. Recently, advanced research has also been developed to reduce the computational complexity of a direct matrix solution [30]. However, not solving a matrix always has its computational advantages as compared to solving a matrix.

We have also compared the accuracy between the two methods using the analytical data as the reference, since the example is set up to have an analytical solution. The entire solution error of the proposed method measured by (42) is shown to be less than 4×10^{-4} across the entire time window. The entire

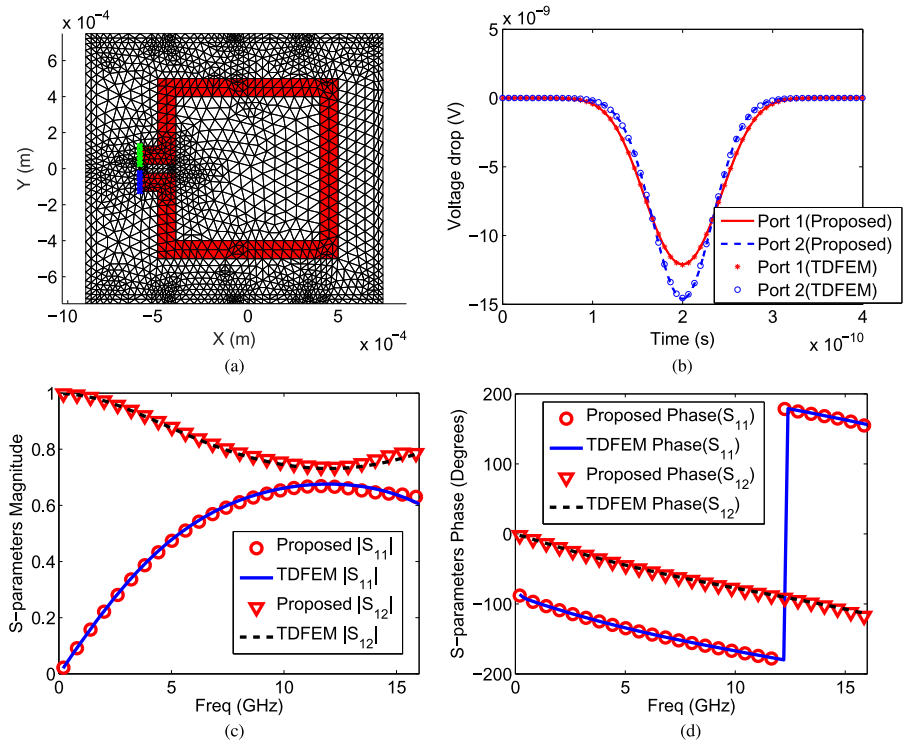


Fig. 14. Simulation of a 3-D package inductor with dielectrics and lossy conductors. (a) Top view of the triangular prism element mesh. (b) Time-domain voltages at the two ports. (c) Magnitude of S-parameters. (d) Phase of S-parameters.

solution error of the TDFEM is shown to be less than 10^{-4}. The accuracy of the proposed method is satisfactory. Meanwhile, the slightly better accuracy of the Galerkin-based TDFEM could be attributed to the fact that it satisfies the Maxwell's equations in an integration sense across each element, whereas the proposed method let the Maxwell's equations be satisfied only at discrete $\mathbf{E}$ and $\mathbf{H}$ points. Furthermore, in the TDFEM, both Faraday's law and Ampere's law are satisfied in the same element, whereas in the proposed method, the second law is satisfied across the elements over the loops orthogonal to the first field unknowns. In addition, the time discretization scheme may also contribute to the difference in accuracy.

VII. Conclusion

In this paper, a new matrix-free time-domain method with a naturally diagonal mass matrix is developed for solving Maxwell's equations in 3-D unstructured meshes, whose accuracy and stability are theoretically guaranteed. Its property of being free of matrix solution is independent of element shape, thus suitable for analyzing arbitrarily shaped structures and materials discretized into unstructured meshes. The method is neither FDTD nor TDFEM, but it possesses the advantage of the FDTD in being naturally matrix free, and the merit of the TDFEM in handling arbitrarily unstructured meshes. No dual mesh, mass-lumping, interpolation, and projection are required. In addition, the framework of the proposed method permits the use of any higher-order vector basis function, thus allowing for any desired higher order of accuracy in both electric and magnetic fields. Moreover, the formulations presented in this paper do not require any modification on the traditional vector bases. Extensive numerical experiments on unstructured triangular, tetrahedral, triangular prism meshes, and mixed elements have validated the accuracy, matrix-free property, stability, and generality of the proposed method. Comparisons have also been made with the TDFEM in unstructured meshes in CPU time, memory consumption, and accuracy, which demonstrate the merits of the proposed method.

Appendix
First-Order Curl-Conforming Vector Basis Functions

In a tetrahedral element, among the 20 first-order vector bases [26], there are 12 edge vector basis functions, which are defined as

$$\begin{aligned}
\mathbf{N}_1 &= (3\xi_2 - 1)\mathbf{W}_1, \quad \mathbf{N}_2 = (3\xi_1 - 1)\mathbf{W}_1 \\
\mathbf{N}_3 &= (3\xi_1 - 1)\mathbf{W}_2, \quad \mathbf{N}_4 = (3\xi_3 - 1)\mathbf{W}_2 \\
\mathbf{N}_5 &= (3\xi_4 - 1)\mathbf{W}_3, \quad \mathbf{N}_6 = (3\xi_1 - 1)\mathbf{W}_3 \\
\mathbf{N}_7 &= (3\xi_3 - 1)\mathbf{W}_4, \quad \mathbf{N}_8 = (3\xi_2 - 1)\mathbf{W}_4 \\
\mathbf{N}_9 &= (3\xi_2 - 1)\mathbf{W}_5, \quad \mathbf{N}_{10} = (3\xi_4 - 1)\mathbf{W}_5 \\
\mathbf{N}_{11} &= (3\xi_4 - 1)\mathbf{W}_6, \quad \mathbf{N}_{12} = (3\xi_3 - 1)\mathbf{W}_6
\end{aligned} \tag{44}$$

where ξ_i $(i = 1, 2, 3, 4)$ are volume coordinates, and $\mathbf{W}_i$ $(i = 1, 2, \ldots, 6)$ denote the normalized zeroth-order edge bases as follows:

$$\begin{aligned}
\mathbf{W}_1 &= L_1(\xi_2\nabla\xi_1 - \xi_1\nabla\xi_2) \\
\mathbf{W}_2 &= L_2(\xi_1\nabla\xi_3 - \xi_3\nabla\xi_1) \\
\mathbf{W}_3 &= L_3(\xi_4\nabla\xi_1 - \xi_1\nabla\xi_4) \\
\mathbf{W}_4 &= L_4(\xi_3\nabla\xi_2 - \xi_2\nabla\xi_3) \\
\mathbf{W}_5 &= L_5(\xi_2\nabla\xi_4 - \xi_4\nabla\xi_2) \\
\mathbf{W}_6 &= L_6(\xi_4\nabla\xi_3 - \xi_3\nabla\xi_4)
\end{aligned} \tag{45}$$

in which L_i is the length of the ith edge. The degrees of freedom of the 12 edge vector bases shown in (44) are located respec-

tively at the following points in each element, with their corresponding projection directions $\hat{e}_i$ $(i = 1, 2, \ldots, 12)$ defined as:

$$\begin{aligned}
\hat{e}_1 &= \frac{\vec{v}_{21}}{\|\vec{v}_{21}\|},\ \mathbf{r}_{e1} = \left(\xi_1 = \frac{1}{3}, \xi_2 = \frac{2}{3}, \xi_3 = 0, \xi_4 = 0\right)\\
\hat{e}_2 &= \hat{e}_1,\ \mathbf{r}_{e2} = \left(\xi_1 = \frac{2}{3}, \xi_2 = \frac{1}{3}, \xi_3 = 0, \xi_4 = 0\right)\\
\hat{e}_3 &= \frac{\vec{v}_{13}}{\|\vec{v}_{13}\|},\ \mathbf{r}_{e3} = \left(\xi_1 = \frac{2}{3}, \xi_2 = 0, \xi_3 = \frac{1}{3}, \xi_4 = 0\right)\\
\hat{e}_4 &= \hat{e}_3,\ \mathbf{r}_{e4} = \left(\xi_1 = \frac{1}{3}, \xi_2 = 0, \xi_3 = \frac{2}{3}, \xi_4 = 0\right)\\
\hat{e}_5 &= \frac{\vec{v}_{41}}{\|\vec{v}_{41}\|},\ \mathbf{r}_{e5} = \left(\xi_1 = \frac{1}{3}, \xi_2 = 0, \xi_3 = 0, \xi_4 = \frac{2}{3}\right)\\
\hat{e}_6 &= \hat{e}_5,\ \mathbf{r}_{e6} = \left(\xi_1 = \frac{2}{3}, \xi_2 = 0, \xi_3 = 0, \xi_4 = \frac{1}{3}\right)\\
\hat{e}_7 &= \frac{\vec{v}_{32}}{\|\vec{v}_{32}\|},\ \mathbf{r}_{e7} = \left(\xi_1 = 0, \xi_2 = \frac{1}{3}, \xi_3 = \frac{2}{3}, \xi_4 = 0\right)\\
\hat{e}_8 &= \hat{e}_7,\ \mathbf{r}_{e8} = \left(\xi_1 = 0, \xi_2 = \frac{2}{3}, \xi_3 = \frac{1}{3}, \xi_4 = 0\right)\\
\hat{e}_9 &= \frac{\vec{v}_{24}}{\|\vec{v}_{24}\|},\ \mathbf{r}_{e9} = \left(\xi_1 = 0, \xi_2 = \frac{2}{3}, \xi_3 = 0, \xi_4 = \frac{1}{3}\right)\\
\hat{e}_{10} &= \hat{e}_9,\ \mathbf{r}_{e10} = \left(\xi_1 = 0, \xi_2 = \frac{1}{3}, \xi_3 = 0, \xi_4 = \frac{2}{3}\right)\\
\hat{e}_{11} &= \frac{\vec{v}_{43}}{\|\vec{v}_{43}\|},\ \mathbf{r}_{e11} = \left(\xi_1 = 0, \xi_2 = 0, \xi_3 = \frac{1}{3}, \xi_4 = \frac{2}{3}\right)\\
\hat{e}_{12} &= \hat{e}_{11},\ \mathbf{r}_{e12} = \left(\xi_1 = 0, \xi_2 = 0, \xi_3 = \frac{2}{3}, \xi_4 = \frac{1}{3}\right)
\end{aligned} \tag{46}$$

where $\vec{v}_{ij}$ denotes the vector pointing from node i to node j.

There are also two vector basis functions whose degrees of freedom are located at the center point of each face. In total, there are eight such bases, which are

$$\begin{aligned}
\mathbf{N}_{13} &= 4.5\xi_2\mathbf{W}_6, \quad \mathbf{N}_{14} = 4.5\xi_3\mathbf{W}_5\\
\mathbf{N}_{15} &= 4.5\xi_3\mathbf{W}_3, \quad \mathbf{N}_{16} = 4.5\xi_4\mathbf{W}_2\\
\mathbf{N}_{17} &= 4.5\xi_4\mathbf{W}_1, \quad \mathbf{N}_{18} = 4.5\xi_1\mathbf{W}_5\\
\mathbf{N}_{19} &= 4.5\xi_1\mathbf{W}_4, \quad \mathbf{N}_{20} = 4.5\xi_2\mathbf{W}_2.
\end{aligned} \tag{47}$$

The locations $\mathbf{r}_{ei}(i = 13, 14, \ldots, 20)$ and corresponding unit vectors $\hat{e}_i$ associated with the above eight face vector bases are

$$\begin{aligned}
\hat{e}_{13} &= \hat{e}_{11}, \quad \mathbf{r}_{13} = \left(\xi_2 = \xi_3 = \xi_4 = \frac{1}{3}, \xi_1 = 0\right)\\
\hat{e}_{14} &= \hat{e}_9, \quad \mathbf{r}_{14} = \left(\xi_2 = \xi_3 = \xi_4 = \frac{1}{3}, \xi_1 = 0\right)\\
\hat{e}_{15} &= \hat{e}_5, \quad \mathbf{r}_{15} = \left(\xi_1 = \xi_3 = \xi_4 = \frac{1}{3}, \xi_2 = 0\right)\\
\hat{e}_{16} &= \hat{e}_3, \quad \mathbf{r}_{16} = \left(\xi_1 = \xi_3 = \xi_4 = \frac{1}{3}, \xi_2 = 0\right)\\
\hat{e}_{17} &= \hat{e}_1, \quad \mathbf{r}_{17} = \left(\xi_1 = \xi_2 = \xi_4 = \frac{1}{3}, \xi_3 = 0\right)\\
\hat{e}_{18} &= \hat{e}_9, \quad \mathbf{r}_{18} = \left(\xi_1 = \xi_2 = \xi_4 = \frac{1}{3}, \xi_3 = 0\right)\\
\hat{e}_{19} &= \hat{e}_7, \quad \mathbf{r}_{19} = \left(\xi_1 = \xi_2 = \xi_3 = \frac{1}{3}, \xi_4 = 0\right)\\
\hat{e}_{20} &= \hat{e}_3, \quad \mathbf{r}_{20} = \left(\xi_1 = \xi_2 = \xi_3 = \frac{1}{3}, \xi_4 = 0\right).
\end{aligned} \tag{48}$$

REFERENCES

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Trans. Antennas Propag.*, vol. AP-14, no. 3, pp. 302–307, May 1966.

[2] A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Boston, MA, USA: Artech House, 2000.

[3] R. Holland, "Finite-difference solution of maxwell's equations in generalized nonorthogonal coordinates," *IEEE Trans. Nucl. Sci.*, vol. NS-30, no. 6, pp. 4589–4591, Dec. 1983.

[4] M. Fusco, "FDTD algorithm in curvilinear coordinates [EM scattering]," *IEEE Trans. Antennas Propag.*, vol. 38, no. 1, pp. 76–89, Jan. 1990.

[5] J.-F. Lee, R. Palandech, and R. Mittra, "Modeling three-dimensional discontinuities in waveguides using nonorthogonal FDTD algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 40, no. 2, pp. 346–352, Feb. 1992.

[6] T. G. Jurgens and A. Taflove, "Three-dimensional contour FDTD modeling of scattering from single and multiple bodies," *IEEE Trans. Antennas Propag.*, vol. 41, no. 12, pp. 1703–1708, Dec. 1993.

[7] S. Dey and R. Mittra, "A locally conformal finite difference time domain (FDTD) algorithm for modeling three-dimensional perfectly conducting objects," *IEEE Microw. Guided Wave Lett.*, vol. 7, no. 9, pp. 273–275, 1997.

[8] Y. Hao and C. J. Railton, "Analyzing electromagnetic structures with curved boundaries on Cartesian FDTD meshes," *IEEE Trans. Microw. Theory Techn.*, vol. 46, no. 1, pp. 82–88, Jan. 1998.

[9] M. Madsen, "Divergence preserving discrete surface integral methods for maxwells equations using nonorthogonal grids," *J. Comput. Phys.*, vol. 119, pp. 34–45, 1995.

[10] C. Chan, J. Elson, and H. Sangani, "An explicit finite-difference time-domain method using whitney elements," in *Proc. IEEE Int. Symp. Antennas Propag. (AP-S)*, 1994, vol. 3, pp. 1768–1771.

[11] S. Gedney, F. Lansing, and D. Rascoe, "A full-wave analysis of passive monolithic integrated circuit devices using a generalized yee-algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 44, no. 8, pp. 1393–1400, Aug. 1996.

[12] A. Bossavit and L. Kettunen, "Yee-like schemes on a tetrahedral mesh, with diagonal lumping," *Int. J. Numer. Modelling-Electron. Networks Devices Fields*, vol. 12, no. 1, pp. 129–142, 1999.

[13] C. F. Lee, B. J. McCartin, R. T. Shin, and J. A. Kong, "A triangle grid finite-difference time-domain method for electromagnetic scattering problems," *J. Electromagn. Waves Appl.*, vol. 8, no. 4, pp. 1429–1438, Aug. 1994.

[14] M. Hano and T. Itoh, "Three-dimensional time-domain method for solving maxwells equations based on circumcenters of elements," *IEEE Trans. Magn.*, vol. 32, no. 3, pp. 946–949, May 1996.

[15] S. Gedney and J. Roden, "Numerical stability of nonorthogonal fdtd methods," *IEEE Trans. Antennas Propag.*, vol. 48, no. 2, pp. 231–239, Feb. 2000.

[16] M. Cinalli and A. Schiavoni, "A stable and consistent generalization of the fdtd technique to nonorthogonal unstructured grids," *IEEE Trans. Antennas Propag.*, vol. 54, no. 5, pp. 1503–1512, May 2006.

[17] D. Jiao and J. Jin, "Finite element analysis in time domain," in *The Finite Element Method in Electromagnetics*. Hoboken, NJ, USA: Wiley, 2002, pp. 529–584.

[18] M. Feliziani and F. Maradei, "Hybrid finite-element solutions as time dependent maxwells curl equations," *IEEE Trans. Magn.*, vol. 31, no. 3, pp. 1330–1335, May 1995.

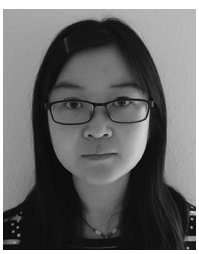
[19] D. A. White, "Orthogonal vector basis functions for time domain finite element solution of the vector wave equation," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1458–1461, May 1999.

[20] D. Jiao and J. Jin, "Three-dimensional orthogonal vector basis functions for time-domain finite element solution of vector wave equations," *IEEE Trans. Antennas Propag.*, vol. 51, no. 1, pp. 59–66, Jan. 2003.

[21] S. D. Gedney *et al.*, "The discontinuous galerkin finite element time domain method (DGFETD)," in *Proc. IEEE Int. Symp. Antennas Propag.*, 2008, p. 4.

[22] S. D. Gedney, J. C. Young, T. C. Kramer, and J. A. Roden, "A discontinuous galerkin finite element time-domain method modeling of dispersive media," *IEEE Trans. Antennas Propag.*, vol. 60, no. 4, pp. 1969–1977, Apr. 2012.

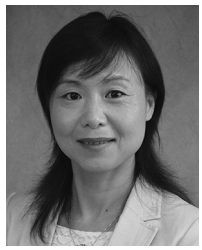
[23] J. Yan and D. Jiao, "A matrix-free time-domain method independent of element shape for electromagnetic analysis," in *Proc. IEEE Int. Symp. Antennas Propag. (AP-S)*, 2014, pp. 2258–2259.
[24] J. Yan and D. Jiao, "Accurate matrix-free time-domain method in unstructured meshes," in *Proc. IEEE Int. Microw. Symp. (IMS)*, 2015, pp. 1–4.
[25] J. Jin, *The Finite Element Method in Electromagnetics*. Hoboken, NJ, USA: Wiley, 2014.
[26] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 45, no. 3, pp. 329–342, Mar. 1997.
[27] D. Jiao and J. M. Jin, "A general approach for the stability analysis of time-domain finite element method," *IEEE Trans. Antennas Propag.*, vol. 50, no. 11, pp. 1624–1632, Nov. 2002.
[28] Q. He, H. Gan, and D. Jiao, "Explicit time-domain finite-element method stabilized for an arbitrarily large time step," *IEEE Trans. Antennas Propag.*, vol. 60, no. 11, pp. 5240–5250, Nov. 2012.
[29] M.-F. Wong, O. Picon, and V. F. Hanna, "A finite element method based on Whitney forms to solve maxwells equations in the time domain," *IEEE Trans. Magn.*, vol. 31, no. 3, pp. 1618–1621, May 1995.
[30] B. Zhou and D. Jiao, "Direct finite-element solver of linear complexity for large-scale 3-d electromagnetic analysis and circuit extraction," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 10, pp. 3066–3080, Oct. 2015.



Jin Yan received the B.S. degree in electronic engineering and information science from the University of Science and Technology of China, Hefei, China, in 2012. She is currently working toward the Ph.D. degree in electrical engineering at Purdue University, West Lafayette, IN, USA.

She currently works in the On-Chip Electromagnetics Group at Purdue University. Her research is focused on computational electromagnetics, high-performance VLSI CAD, and fast and high-capacity numerical methods.

Ms. Yan was the recipient of an Honorable Mention Award of the IEEE International Symposium on Antennas and Propagation in 2015.



Dan Jiao (S'00–M'02–SM'06) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 2001.

She then joined the Technology Computer-Aided Design (CAD) Division, Intel Corporation, until September 2005, where she was a Senior CAD Engineer, Staff Engineer, and Senior Staff Engineer. In September 2005, she joined Purdue University, West Lafayette, IN, USA, as an Assistant Professor with the School of Electrical and Computer Engineering. She is currently a Professor with Purdue University. She has authored 3 book chapters and over 230 papers in refereed journals and international conferences. Her current research interests include computational electromagnetics; high-frequency digital, analog, mixed-signal, and RF integrated circuit (IC) 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, and millimeter-wave circuits; wireless communication; and bio-electromagnetics.

Dr. Jiao has served as the reviewer for many IEEE journals and conferences. She is an Associate Editor of the IEEE TRANS. ON COMPONENTS, Packaging, and Manufacturing Technology. She received 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 ELATE (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 US Frontiers of Engineering Symposium. She was the recipient of the 2010 Ruth and Joel Spira Outstanding Teaching Award, the 2008 National Science Foundation (NSF) CAREER Award, the 2006 Jack and Cathie Kozik Faculty Start up Award (which recognizes an outstanding new faculty member of the School of Electrical and Computer Engineering, Purdue University), a 2006 Office of Naval Research (ONR) Award under the Young Investigator Program, the 2004 Best Paper Award presented at the Intel Corporations annual corporate-wide technology conference (Design and Test Technology Conference) for her work on generic broadband model of high-speed circuits, the 2003 Intel Corporations Logic Technology Development (LTD) Divisional Achievement Award, the Intel Corporations Technology CAD Divisional Achievement Award, the 2002 Intel Corporations Components Research the Intel Hero Award (Intel-wide she was the tenth recipient), the Intel Corporations LTD Team Quality Award, and the 2000 Raj Mittra Outstanding Research Award presented by the University of Illinois at Urbana-Champaign.