12-2016

# Matrix-free time-domain methods for general electromagnetic analysis

Jin Yan
*Purdue University*

*Graduate School Form*
*30 Updated 12/26/2015*

**PURDUE UNIVERSITY**
**GRADUATE SCHOOL**
**Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By JIN YAN

Entitled

MATRIX-FREE TIME-DOMAIN METHODS FOR GENERAL ELECTROMAGNETIC ANALYSIS

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

DAN JIAO
Chair

DIMITRIOS PEROULIS

KEVIN J. WEBB

CHENG-KOK KOH

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): DAN JIAO

Approved by: VENKATARAMANAN BALAKRISHNAN 12/5/2016

Head of the Departmental Graduate Program Date

# MATRIX-FREE TIME-DOMAIN METHODS FOR GENERAL ELECTROMAGNETIC ANALYSIS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Jin Yan

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2016

Purdue University

West Lafayette, Indiana

To my family

# ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Professor Dan Jiao for giving me this opportunity to work with her at Purdue University. She introduced me to this research area and guided me step by step to tackle those challenging research problems in an elegant and simple way. Through the past four years, she is always there whenever I need a discussion, and she never gives up exploring unless the root cause is found. This thesis can never be completed without her sharp insights, constant inspiration and encouragement.

I would like to thank other members of my PhD advisory committee: Professor Cheng-Kok Koh, Professor Kevin J. Webb and Professor Dimitrios Peroulis for their precious time, support and suggestions regarding my research work.

I would also like to express my gratitude to Dr. Dzevat Omeragic and Dr. Michael Thiel at the Schlumberger-Doll Research Center (SDR) for giving me an opportunity to work as an internee in the Mathematical and Modeling Group. This internship experience really broadened my horizons and helped me in polishing my research skills.

Thanks are also extended to all the past and current members of On-Chip Electromagnetics Lab here at Purdue: Dr. Houle Gan, Dr. Jongwon Lee, Dr. Jianfang Zhu, Dr. Wenwen Chai, Dr. Duo Chen, Dr. Haixing Liu, Dr. Feng Sheng, Dr. Qing He, Dr. Saad Omar, Dr. Bangda Zhou, Dr. Md Gaffar, Dr. Woochan Lee, Dr. Yanpu Zhao, Dr. Ping Li, Miaomiao Ma, Li Xue, Kaiyuan Zeng, Chang Yang and Yu Zhao for their constant support and the friendly yet professional research-conducive environment in the lab.

Last but not the least, I am thoroughly indebted to my parents, my younger sister and my husband for their everlasting love and support during all these years. I dedicate this work to all of you!

TABLE OF CONTENTS

Page

Page

Page

Page

# LIST OF TABLES

# LIST OF FIGURES

Figure Page

Figure Page

# ABSTRACT

Yan, Jin Ph.D., Purdue University, December 2016. Matrix-Free Time-Domain Methods for General Electromagnetic Analysis. Major Professor: Dan Jiao.

Many engineering challenges demand an efficient computational solution of large-scale problems. If a computational method can be made 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 is free of matrix solutions. However, it requires a structured orthogonal grid for space discretization. In this work, we develop a new time-domain method that naturally requires no matrix solution, regardless of whether the discretization is a structured grid or an unstructured mesh. No dual mesh, interpolation, projection and mass lumping are needed. Furthermore, a time-marching scheme is developed to ensure the stability for simulating an unsymmetrical numerical system, while preserving the matrix-free merit of the proposed method. This time-marching scheme is then made unconditionally stable, and hence allowing for the use of an arbitrarily large time step without sacrificing the matrix-free property. Extensive numerical experiments have been carried out on a variety of two- and three-dimensional unstructured meshes and even mixed-element meshes. Correlations with analytical solutions and the results obtained from the time-domain finite-element method have validated the accuracy, matrix-free property, stability, and generality of the proposed method.

In addition to an extensive development of the proposed method in arbitrary 2- and 3-D unstructured meshes, we have also made a connection between the proposed new method and the classical FDTD method. We have found that the proposed matrix-free method naturally reduces to the FDTD method in an orthogonal grid. It also results in a new patch-based single-grid formulation of the FDTD algorithm.

This new formulation not only makes the implementation of the original FDTD much easier, but also reveals a natural rank-1 decomposition of the curl-curl operator. Such a representation leads to an efficient extraction of unstable eigenmodes from fine cells only, from which a fast explicit and unconditionally stable FDTD method is developed. In addition, to efficiently handle multiscale structures, we develop an accurate FDTD subgridding algorithm suitable for arbitrary subgridding settings with arbitrary contrast ratios between the normal gird and the subgrid. Although the resulting system matrix is unsymmetric, we develop a time marching method to overcome the stability problem without sacrificing the matrix-free merit of the original FDTD. This method is general, which is also applicable to other subgridding algorithms whose underlying numerical systems are unsymmetric. The proposed FDTD subgridding algorithm is then further made unconditionally stable, thus permitting the use of a time step independent of space step.

Last but not the least, the framework of the proposed method can be flexibly extended to solve partial differential equations in other disciplines, which we have demonstrated for thermal analysis.

# 1. INTRODUCTION

## 1.1 Background and Motivation

To tackle the real-world challenges in science and engineering, a computational solution is demanded to solve very large-scale problems. If a computational method can be made matrix-free, i.e., free of matrix solutions, then it has a potential to solve much larger problems.

Among existing computational electromagnetic methods, the explicit finite-difference time-domain (FDTD) method [1,2] is free of matrix solutions. However, its time step is restricted by space step. To overcome the aforementioned barrier, researchers have developed implicit unconditionally stable FDTD methods, such as the alternating-direction implicit (ADI) method [3,4], the Crank-Nicolson (CN) method [5], the CN-based split step (SS) scheme [6], the pseudo-spectral time-domain (PSTD method) [7], the locally one-dimensional (LOD) FDTD [8,9], the Laguerre FDTD method [10,11], the associated Hermite (AH) type FDTD [12], a series of fundamental schemes [13] and many others, but the advantage of the conventional FDTD is sacrificed in avoiding a matrix solution. When the problem size is large, the implicit unconditionally stable FDTD methods become inefficient. Research has also been pursued to address the time step problem in the original explicit time-domain methods [14–16]. In [17, 18], the source of instability is identified, and subsequently eradicated from the underlying numerical system to make an explicit FDTD unconditionally stable. It is shown that the source of instability is the eigenmodes of the discretized curl-curl operator whose eigenvalues are the largest. These eigenvalues are higher than what can be stably simulated by the given time step. To find these unstable modes, in [18], a partial solution of a global eigenvalue solution is computed. In general, only a small set of the largest eigenpairs of the system matrix need to be found, and the

system matrix is also sparse. However, the computational overhead of the resultant scheme may still be too high to tolerate when the matrix size is large. Another line of thought to solve this problem is to create a subgridding algorithm that locally refines the mesh in the regions where a higher resolution is necessary, thus the number of unknowns to be solved will be reduced. In literature, many FDTD subgridding methods have been proposed from different perspectives, such as variable step size method [19], mesh refinement algorithm (MRA) [20], multigrid displacement method (MGDM) [21], multigrid current method (MGCM) [22] and many others. Although the accuracy of most of these methods is preserved, they all lack a theoretical proof on their stability. As a result, the efficiency and stability of the FDTD method needs to be further improved when fine features exist in the computational domain.

Except for the time step limitation, the FDTD method also requires a structured orthogonal grid for space discretization. To overcome this limitation, many non-orthogonal FDTD methods have been developed such as the curvilinear FDTD [23–25], contour and conformal FDTD [26–28], discrete surface integral (DSI) methods [29], generalized Yee-algorithms [30–35], finite integration technique with affine theories [36], etc. Needless to say, they have significantly advanced the capability of the original FDTD method in handling unstructured meshes. In existing non-orthogonal 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 non-orthogonal 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 **E** is, in general, not reciprocal to that on **H** in existing methods developed for irregular meshes. It can be proved that such a non-reciprocal 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 [35]. 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) [37] 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 technique has been used to diagonalize the mass matrix in TDFEM, and also finite integration technique [36]. But it requires well-shaped elements to be accurate [38]. In addition to mass lumping, orthogonal vector basis functions have been developed to render the mass matrix diagonal [39, 40]. These bases are element-shape dependent. They also rely on an approximate integration to make the mass matrix diagonal. In recent years, Discontinuous Galerkin time-domain methods [41, 42] have been developed, which only involve the solution of local matrices of a small size. However, this is achieved by not enforcing the tangential continuity of the fields across the element interface at each time instant. Certainly, an accurate result would still have to satisfy the continuity conditions of the fields. Not directly 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.

## 1.2 Contribution of This Work

In this work, we provide solutions to the two problems raised above: how to create a matrix-free time-domain method in unstructured meshes, and how to overcome the remaining barriers of a matrix-free method in an orthogonal grid like the FDTD in stability and efficiency when fine spatial features are present?

First, 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 essential idea of the proposed method is to use higher-order vector bases to represent one field unknown in each element, as a result, the other field unknown can be obtained accurately at any point along any direction, without any need for interpolation and projection. Hence, the other field unknown can be sampled in such a way that the first field unknown can be reversely generated with guaranteed accuracy. The resultant mass matrix is naturally diagonal. In addition to ensuring accuracy, we realize that the other key to enable a matrix-free method in an unstructured mesh is to be able to stably simulate an unsymmetrical numerical system. An unsymmetrical operator is often unavoidable in order to ensure the accuracy of a matrix-free discretization of Maxwell's equations in an unstructured mesh. However, it may yield complex-valued and even negative eigenvalues in nature, which makes a traditional explicit marching absolutely unstable. As long as we are able to stably handle complex-valued and also negative eigenvalues, we can fully benefit from the accuracy and flexibility offered by an unsymmetrical operator in unstructured meshes. This algorithm is developed in this work, without sacrificing the merit of being free of a matrix solution.

The proposed matrix-free time-domain method is further made unconditionally stable with very little cost such that it permits the usage of any large time step irrespective of space step. Therefore, the maximum time step that can be used by the proposed method is no longer restricted by the smallest space step in the mesh. Meanwhile, if the given time step is chosen based on accuracy, then the accuracy of the proposed method is also guaranteed.

Next, we propose a fast and explicit unconditionally stable FDTD method without a global eigenvalue solution. First of all, a new patch-based single-grid FDTD formulation is developed, by using which, we identify the theoretical relationship between fine cells and the largest eigenmodes of the underlying system matrix. We prove that once there exists a difference between the time step required by stability and the time step determined by accuracy, i.e., a difference between the fine-cell size and the regular-cell size, the largest eigenmodes of the original system matrix can be extracted from fine cells. The larger the contrast ratio between the two time steps, the more accurate the eigenmodes extracted in this way. Based on this theoretical finding, we propose an efficient algorithm to find the unstable modes directly from fine cells, and subsequently deduct these unstable modes from the numerical system to achieve an explicit time marching with unconditional stability.

To efficiently handle fine features as well as multiscale structures, subgridding has been used to locally refine the grid in an FDTD simulation. To preserve accuracy in a grid with arbitrary subgrids, an FDTD subgridding scheme, in general, would result in an unsymmetric numerical system. Such a numerical system can have complex-valued eigenvalues, which will render a traditional explicit time marching of FDTD absolutely unstable. In this work, we develop an accurate FDTD subgridding algorithm suitable for arbitrary subgridding settings with arbitrary contrast ratios between the normal gird and the subgrid. Although the resulting system matrix is also unsymmetric, we develop a time marching method to overcome the stability problem without sacrificing the matrix-free merit of the original FDTD. This method is general, which is also applicable to other subgridding algorithms whose underly-

ing numerical systems are unsymmetric. The proposed FDTD subgridding algorithm is then further made unconditionally stable, thus permitting the use of a time step independent of space step.

Last but not the least, we have also shown that the proposed matrix-free method has a potential to solve general parietal differential equations. Hence, its usage is not limited to just the solution of Maxwell's equations, as evidenced by our successful simulations of thermal problems.

### 1.3 Dissertation Outline

The remainder of this dissertation is organized as follows.

In Chap. 2, we develop a new time-domain method that requires no matrix solution, regardless of whether the discretization is a structured grid or an unstructured mesh. We first introduce a general framework for creating a matrix-free time-domain method, then present the detailed formulations for 2-D problems, including the modification of traditional vector bases and the choices of sampled H-points and H-directions. A new time marching scheme is introduced to ensure stability meanwhile preserve matrix-free property. In addition, a comprehensive analysis is conducted on the accuracy and stability of the proposed method. Numerical experiments have been conducted on a variety of unstructured meshes. Correlations with analytical solutions and the time-domain finite-element method that is capable of handling unstructured meshes have validated the accuracy and generality of the proposed matrix-free method.

In Chap. 3, we develop a matrix-free time-domain method for simulating 3-D structures under the same framework as is described in Chap. 2. How to modify the traditional vector basis functions for both tetrahedral element and triangular prism element is presented in details. The validity of the modification on traditional vector basis functions is also explained. In numerical results section, we simulate a variety

of 3-D unstructured meshes involving inhomogeneous materials and conductors to validate the proposed method.

In Chap. 4, we develop a new matrix-free time-domain method without the need for modifying the traditional vector basis functions. 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. 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 to validate the accuracy, matrix-free property, stability, and generality of the proposed method.

In Chap. 5, we develop an unconditionally stable matrix-free time-domain method for analyzing general electromagnetic problems discretized into arbitrarily shaped unstructured meshes. This method does not require the solution of a system matrix, no matter which element shape is used for space discretization. Furthermore, this property is achieved irrespective of the time step used to perform the time domain simulation. As a result, the time step can be solely determined by accuracy regardless of space step. How the proposed method works is studied theoretically. Moreover, the complexity of the proposed method is also presented. Numerical experiments have validated the accuracy and efficiency of the proposed new method.

In Chap. 6, we first propose a new patch-based single-grid FDTD formulation under the framework of the matrix-free time-domain method introduced in previous chapters. Based on this new formulation, we develop a fast explicit and unconditionally stable FDTD method without global eigenvalue solution. In this method, we find the relationship between the unstable modes and the fine meshes, and use this relationship to directly identify the source of instability. We then upfront eradicate the source of instability from the numerical system before performing an explicit time marching. The resultant simulation is absolutely stable for the given time step

irrespective of how large it is. Numerical experiments have demonstrated a significant speedup of the proposed method over the conventional FDTD method as well as state-of-the-art explicit and unconditionally stable methods.

In Chap. 7, we develop an accurate FDTD subgridding algorithm suitable for arbitrary subgridding settings with arbitrary contrast ratios between the normal gird and the subgrid. Although the resulting system matrix is unsymmetric, which makes the traditional explicit time marching definitely unstable, we develop a time marching method to overcome the stability problem without sacrificing the matrix-free merit of the original FDTD. This method is general, which is also applicable to other subgridding algorithms whose underlying numerical systems are unsymmetric. The proposed FDTD subgridding algorithm is then further made unconditionally stable, thus permitting the use of a time step independent of space step. Extensive numerical experiments involving both 2- and 3-D subgrids with various contrast ratios have demonstrated the accuracy, stability, and efficiency of the proposed subgridding algorithm.

In Chap. 8, to demonstrate the generality of the proposed matrix-free method for solving other PDEs, we apply the method to solve thermal diffusion equations. By appending the temperature with a direction and introducing an auxiliary variable, the scalar thermal diffusion equation has been transformed into two vector equations to solve using the matrix-free time-domain method. The effectiveness of the proposed method has been validated by numerical experiments in both time and frequency domain.

In Chap. 9, we summarize the work that has been done and also present our future work.

# 2. MATRIX-FREE TIME-DOMAIN METHOD IN 2-D UNSTRUCTURED MESHES

## 2.1 Introduction

In this chapter, we present a new time-domain method that has a naturally diagonal mass matrix and thereby a strict linear computational complexity per time step, regardless of whether the discretization is a structured grid or an unstructured mesh. This property is obtained independent of the element shape used for discretization. No interpolations, projections, and mass lumping are required. The accuracy and stability of the proposed method are theoretically analyzed and shown to be guaranteed. In addition, no dual mesh is needed and the tangential continuity of the fields is satisfied across the element interface. The flexible framework of the proposed method also allows for a straightforward extension to higher-order accuracy in both electric and magnetic fields. Numerical experiments have been conducted on a variety of unstructured meshes. Correlations with analytical solutions and the time-domain finite-element method that is capable of handling unstructured meshes have validated the accuracy and generality of the proposed matrix-free method. This method is also extended to simulate 3-D structures in next chapter.

## 2.2 Proposed Framework

Consider a general electromagnetic problem discretized into arbitrarily shaped elements, which can also be a mix of different kinds of elements such as a mix of brick, triangular prism, and tetrahedral elements. Starting from the differential form of Faraday's law and Ampere's law,

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \tag{2.1}$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E} + \mathbf{J}, \tag{2.2}$$

we pursue a discretization of the above two equations, which results in a numerical system having a diagonal mass matrix in nature. Notice that the other two Maxwell's equations are implicitly satisfied by (2.1) and (2.2).

To discretize Faraday's law (2.1), we expand the electric field $\mathbf{E}$ in each element by certain vector basis functions $\mathbf{N}_i$ $(i = 1, 2, ..., m)$ as the following

$$\mathbf{E} = \sum_{j=1}^{m} e_j \mathbf{N}_j, \tag{2.3}$$

where $e_j$ is the unknown coefficient of the $j$-th vector basis. Using (2.3) and (2.1), we can obtain magnetic field $\mathbf{H}$ at any point. Assume that we compute $\mathbf{H}$ at $N_h$ discrete points, each of which is denoted by $\mathbf{r}_{hi}(i = 1, 2, ..., N_h)$. At each $\mathbf{H}$-point, assume the unit vector along which we compute $\mathbf{H}$ is $\hat{h}_i$. Substituting (2.3) into (2.1), evaluating $\mathbf{H}$ at the $N_h$ points, and taking the dot product of the resultant with corresponding $\hat{h}_i$ at each point, we obtain the following $N_h$ equations:

$$\hat{h}_i \cdot \sum e_j \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}) = -\hat{h}_i \cdot \mu(\mathbf{r}_{hi}) \frac{\partial \mathbf{H}(\mathbf{r}_{hi})}{\partial t}, (i = 1, 2, ..., N_h) \tag{2.4}$$

which can further be compactly written as the following matrix equation:

$$\mathbf{S}_e\{e\} = -diag(\{\mu\}) \frac{\partial \{h\}}{\partial t}, \tag{2.5}$$

where $\{e\}$ is a global vector containing the unknown coefficients $e_i$ of $\mathbf{E}$'s vector bases, and $\{h\}$ is a global vector containing discretized $\mathbf{H}$. Their $i$-th entries are

$$e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i \tag{2.6}$$

$$h_i = \mathbf{H}(\mathbf{r}_{hi}) \cdot \hat{h}_i, \tag{2.7}$$

in which $\mathbf{r}_{ei}$ and $\hat{e}_i$ $(i = 1, 2, ..., N_e)$ are, respectively, the points and the unit-vectors associated with the vector $\mathbf{E}$'s degrees of freedom. In (2.5), $diag(\{\mu\})$ is a diagonal matrix of size $N_h$, whose $i$-th diagonal entry is the permeability at point $\mathbf{r}_{hi}$. The sparse matrix $\mathbf{S}_e$ is rectangular of dimension $N_h$ by $N_e$, the length of $\{e\}$ is $N_e$; while that of $\{h\}$ is $N_h$.

To discretize Ampere's law (2.2), we evaluate $\mathbf{E}$ at the $\mathbf{r}_{ei}$ $(i = 1, 2, ..., N_e)$ points, and take the dot product of the resultant with $\hat{e}_i$ at each point, obtaining

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = \epsilon(\mathbf{r}_{ei})\frac{\partial e_i}{\partial t} + \sigma(\mathbf{r}_{ei})e_i + \hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei}), (i = 1, 2, ..., N_e) \tag{2.8}$$

where $\hat{e}_i \cdot \nabla \times \mathbf{H}$ is generated by using $\{h\}$ obtained from (2.5). As a result, we obtain the following discretization of Ampere's law

$$\mathbf{S}_h\{h\} = diag(\{\epsilon\})\frac{\partial\{e\}}{\partial t} + diag(\{\sigma\})\{e\} + \{j\}, \tag{2.9}$$

where the sparse matrix $\mathbf{S}_h$ is of dimension $N_e \times N_h$, and the $i$-th entry of current source vector $\{j\}$ in (2.9) is

$$j_i = \hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei}), \quad (i = 1, 2, ..., N_e). \tag{2.10}$$

In addition, $diag(\{\epsilon\})$ and $diag(\{\sigma\})$ are diagonal, whose $i$-th entry is, respectively, the permittivity and conductivity at point $\mathbf{r}_{ei}$.

A leap-frog based time discretization of (2.5) and (2.9) clearly provides us with a time-marching scheme free of matrix solutions as follows:

$$\{h\}^{n+\frac{1}{2}} = \{h\}^{n-\frac{1}{2}} - diag(\{\frac{1}{\mu}\})\Delta t\mathbf{S}_e\{e\}^n \tag{2.11}$$

$$\left(diag(\{\epsilon\}) + \frac{\Delta t}{2}diag(\{\sigma\})\right)\{e\}^{n+1} = \left(diag(\{\epsilon\}) - \frac{\Delta t}{2}diag(\{\sigma\})\right)\{e\}^n + \Delta t\mathbf{S}_h\{h\}^{n+\frac{1}{2}} - \Delta t\{j\}^n, \tag{2.12}$$

where $\Delta t$ is the time step, and the time instants for $\{e\}$ and $\{h\}$, denoted by superscripts, are staggered by half. Note that neither (2.11) nor (2.12) involves a matrix solution.

The (2.5) and (2.9) can also be solved in a second-order fashion. Taking another time derivative of (2.9) and substituting (2.5), we obtain

$$\frac{\partial^2\{e\}}{\partial t^2} + diag(\{\frac{\sigma}{\epsilon}\})\frac{\partial\{e\}}{\partial t} + \mathbf{S}\{e\} = -diag(\{\frac{1}{\epsilon}\})\frac{\partial\{j\}}{\partial t}, \tag{2.13}$$

where

$$\mathbf{S} = diag(\{\frac{1}{\epsilon}\})\mathbf{S}_h diag(\{\frac{1}{\mu}\})\mathbf{S}_e. \tag{2.14}$$

It is obvious that the above numerical system is also free of matrix solutions with a central-difference based disretization in time. In fact, it can be readily proved that (2.11) and (2.12) are the same as the central-difference based discretization of second-order system (2.13) after eliminating the $\{h\}$-unknown. In addition, the mass matrix shown in (2.13), which is the matrix in front of the second-order time derivative, is obviously diagonal. Hence, no mass lumping is needed. For anisotropic materials whose permittivity and permeability are tensors, the diagonal mass matrix simply becomes a block diagonal matrix whose block size is 3. Hence, its inverse is also explicit, which can be found analytically.

## 2.3 Proposed Formulations

### 2.3.1 General Idea

At this point, it can be seen that the accuracy of the proposed matrix-free method relies on an accurate construction of (2.9) for an arbitrary unstructured mesh, since the accuracy of (2.5) is not a concern at all—with a set of well-established curl-conforming vector basis functions for discretizing **E**, the accuracy of (2.5) is guaranteed for producing **H** at any point and along any direction. Therefore, the key issue is how to build an accurate (2.9). To be more precise, how to construct $\mathbf{S}_h\{h\}$, i.e., a disretization of the curl of **H**, such that it can accurately produce the desired $\{e\}$.

We propose to determine **H** points and directions based on discretized **E** unknowns so that the resultant **H** fields can generate the desired $\{e\}$ accurately. From the integral form of Ampere's law, we know that the circulation of the tangential **H** in a loop can produce an accurate **E** along the direction *normal* to the loop at the *center* point of the loop area. Hence, the simplest approach is for each $\hat{e}_i$ located at point $\mathbf{r}_{ei}$, define a rectangular loop perpendicular to $\hat{e}_i$ and centered at point $\mathbf{r}_{ei}$, as illustrated in Fig. 2.1. Along this loop, we define **H**-points and **H**-directions associated with $\hat{e}_i$. The set of **H**-points and **H**-directions found for each $\hat{e}_i$ at $\mathbf{r}_{ei}$ makes the whole set of **H**-points denoted by $\{\mathbf{r}_{hi}\}$, and the whole set of **H**-directions denoted by $\{\hat{h}_i\}$, with

$(i = 1, 2, ..., N_h)$. The $\{h\}$ is simply the vector of $\mathbf{H}(\mathbf{r}_{hi}) \cdot \hat{h}_i$ $(i = 1, 2, ..., N_h)$ as shown in (2.7). With such an $\{h\}$, the $\mathbf{S}_h$ can be readily built with guaranteed accuracy.

In addition, no dual mesh needs to be constructed for discretizing **H** since the **H** is known from (2.5) at any point and along any direction. We only need to sample **H** at the points along the directions shown in Fig. 2.1 based on **E**'s points and directions. In fact, our discrete **H** does not form a mesh at all.

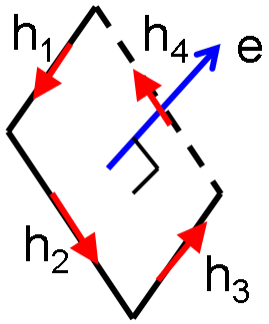



Fig. 2.1. **H** points and directions determined based on **E**'s degrees of freedom.

### 2.3.2 Vector Basis Functions for the Expansion of E

Consider an arbitrary $i$-th edge in a triangular-element based mesh residing on an $x$-$y$ plane. With the normalized zeroth-order edge elements to expand **E**, the $e_i$ shown in (2.6) has $\hat{e}_i$ the unit vector tangential to the $i$-th edge, and $\mathbf{r}_{ei}$ the center point of the $i$-th edge. To obtain such an $e_i$ accurately from the discrete **H** (now $H_z$ only for a 2-D TE case), the two **H**-points should be located on the line that is perpendicular to the $i$-th edge and centered at the point $\mathbf{r}_{ei}$, as illustrated in Fig. 2.2(a). In this way, the edge is perpendicular to the **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. However, using the zeroth-order edge elements, the curl of **E** is constant in every element, thus we cannot generate **H** at the desired points accurately. From another perspective, we can view the **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. 2.2(b).

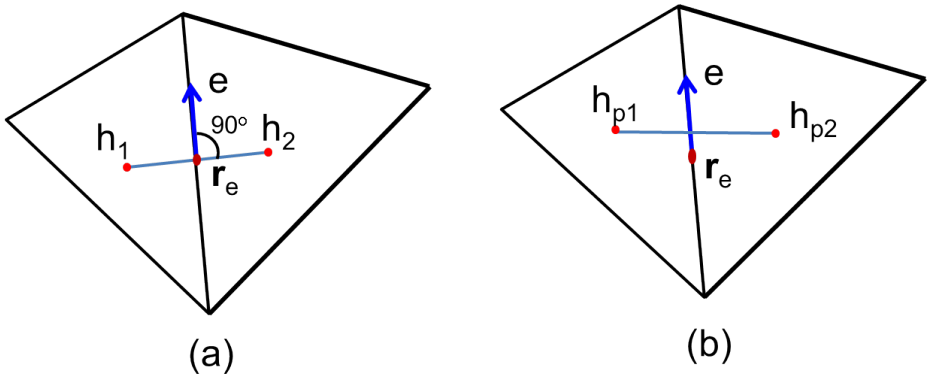



Fig. 2.2. (a) The locations of **H** points required for the accurate evaluation of $e$ at point $\mathbf{r}_e$. (b) The locations of **H** points with zeroth-order edge bases.

To overcome the aforementioned problem, we propose to use a higher-order curl-conforming vector basis to expand **E** in each element. With an order higher than zero, the curl of **E** and hence **H** is at least a linear function in each element. In this way, we can generate **H** at any desired point accurately from (2.5).

However, we cannot blindly use the original set of the first-order curl-conforming vector bases in [43]. They need certain modifications to fit the need of this work. This is because the unknown coefficient $e_i$ shown in (2.3) should be equal to (2.6) to connect (2.5) with (2.9) directly without any need for transformation. This results in the following property of the desired vector basis functions:

$$\begin{aligned} \hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei}) &= 1, \quad j = i \\ \hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei}) &= 0, \quad j \neq i \end{aligned} \tag{2.15}$$

which can be readily obtained by taking a dot product with $\hat{e}_i$ on both sides of (2.3) at point $\mathbf{r}_{ei}$, and recognizing that the left hand side of the resultant is required to be

equal to $e_i$. Notice that (2.15) is not mass lumping that enforces the volume integral of $< \mathbf{N}_i, \mathbf{N}_j >$ to be $\delta_{ij}$.

The zeroth-order edge bases in a triangular or other shaped elements naturally satisfy (2.15). As for the first-order edge basis functions, there are not only six edge degrees of freedom, but also two internal degrees of freedom at the center point of the triangular element. The former six bases satisfy (2.15), but the latter two do not. They hence need a modification. The definitions of these two bases are not unique either, thus they can be modified to satisfy (2.15) without sacrificing the completeness of the bases.

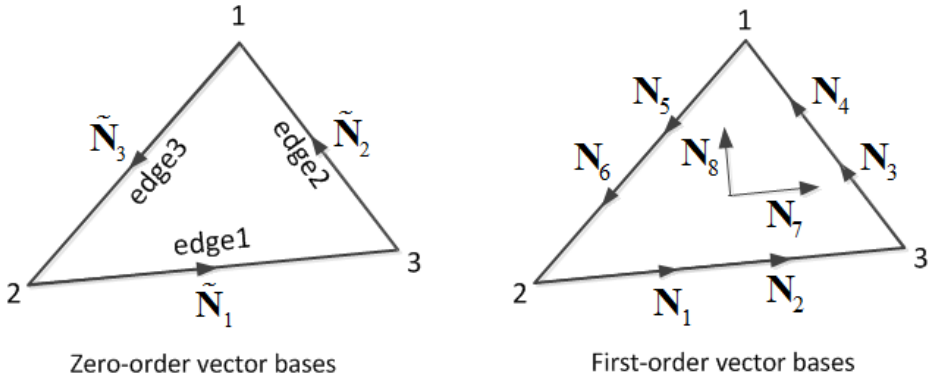



Fig. 2.3. Illustration of the degrees of freedom of the zeroth- and the first-order vector bases in a triangular element.

To elaborate, first, we list the original six edge vector basis functions $\mathbf{N}_i$ $(i = 1, 2, ..., 6)$ together with their unit tangential vectors $\hat{e}_i$ as follows:

$$\begin{aligned}
\hat{e}_1 &= \vec{v}_{23}/\|\vec{v}_{23}\|, \quad \mathbf{N}_1 = (3\xi_2 - 1)\mathbf{W}_1 \\
\hat{e}_2 &= \vec{v}_{23}/\|\vec{v}_{23}\|, \quad \mathbf{N}_2 = (3\xi_3 - 1)\mathbf{W}_1 \\
\hat{e}_3 &= \vec{v}_{31}/\|\vec{v}_{31}\|, \quad \mathbf{N}_3 = (3\xi_3 - 1)\mathbf{W}_2 \\
\hat{e}_4 &= \vec{v}_{31}/\|\vec{v}_{31}\|, \quad \mathbf{N}_4 = (3\xi_1 - 1)\mathbf{W}_2 \\
\hat{e}_5 &= \vec{v}_{12}/\|\vec{v}_{12}\|, \quad \mathbf{N}_5 = (3\xi_1 - 1)\mathbf{W}_3 \\
\hat{e}_6 &= \vec{v}_{12}/\|\vec{v}_{12}\|, \quad \mathbf{N}_6 = (3\xi_2 - 1)\mathbf{W}_3,
\end{aligned} \tag{2.16}$$

where $\vec{v}_{ij}$ denotes the vector pointing from node $i$ to node $j$, as shown in Fig. 2.3, $\xi_i$ $(i = 1, 2, 3)$ are area coordinates, and $\mathbf{W}$ denotes the normalized zeroth-order edge basis as follows

$$\begin{aligned} \mathbf{W}_1 &= L_1(\xi_2 \nabla \xi_3 - \xi_3 \nabla \xi_2) \\ \mathbf{W}_2 &= L_2(\xi_3 \nabla \xi_1 - \xi_1 \nabla \xi_3) \\ \mathbf{W}_3 &= L_3(\xi_1 \nabla \xi_2 - \xi_2 \nabla \xi_1), \end{aligned} \tag{2.17}$$

in which $L_i$ is the length of the $i$-th edge. The degrees of freedom of the above six edge vector bases are located respectively at the following points in each element

$$\begin{aligned} \mathbf{r}_{e1} &= (\xi_2 = 2/3, \xi_3 = 1/3) \\ \mathbf{r}_{e2} &= (\xi_2 = 1/3, \xi_3 = 2/3) \\ \mathbf{r}_{e3} &= (\xi_1 = 1/3, \xi_3 = 2/3) \\ \mathbf{r}_{e4} &= (\xi_1 = 2/3, \xi_3 = 1/3) \\ \mathbf{r}_{e5} &= (\xi_1 = 2/3, \xi_2 = 1/3) \\ \mathbf{r}_{e6} &= (\xi_1 = 1/3, \xi_2 = 2/3). \end{aligned} \tag{2.18}$$

The projection of $\hat{e}_i$ $(i = 1, 2, ..., 6)$ onto any $j$-th vector basis in (2.16) at the point of the $i$-th degree of freedom, i.e. $\hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei})$, is obviously zero for $j \neq i$ and one for $j = i$ . This can be analytically verified, and also conceptually understood because if it is not zero, the first-order bases (2.16) cannot ensure the tangential continuity of $\mathbf{E}$ across the element interfaces, which is not true. Therefore, the property of (2.15) is satisfied for $(i = 1, 2, ..., 6)$ and $(j = 1, 2, ..., 6)$.

For the two vector basis functions whose degrees of freedom are internal at the element center, we have

$$\begin{aligned} \mathbf{r}_{e7} &= (\xi_1 = 1/3, \xi_2 = 1/3) \\ \mathbf{r}_{e8} &= (\xi_1 = 1/3, \xi_2 = 1/3). \end{aligned} \tag{2.19}$$

If we choose the two vector bases as $\mathbf{N}_7 = 4.5\xi_1\mathbf{W}_1$ and $\mathbf{N}_8 = 4.5\xi_2\mathbf{W}_2$ as those suggested in [43], with $\hat{e}_7 = \vec{v}_{23}/\|\vec{v}_{23}\|$ along edge 1, and $\hat{e}_8 = \vec{v}_{31}/\|\vec{v}_{31}\|$ along edge 2, although they make $\hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei})$ zero for $(i = 1, 2, ..., 6)$ and $(j = 7, 8)$, the $\hat{e}_7 \cdot \mathbf{N}_8(\mathbf{r}_{e7})$

is, in general, not zero since edge 1 may not be perpendicular to $\mathbf{W}_2$ at the element center. Thus, (2.15) is not satisfied. If we keep $\mathbf{N}_7$ as it is, but choosing $\mathbf{N}_8$ as $\xi_2\xi_3\nabla\xi_1$, although $\hat{e}_7 \cdot \mathbf{N}_8(\mathbf{r}_{e7})$ becomes zero now, $\hat{e}_8 \cdot \mathbf{N}_7(\mathbf{r}_{e8})$ is not zero in general at the element center. Even though we change $\hat{e}_8$ to be along the direction of $\nabla\xi_1$, $\hat{e}_8 \cdot \mathbf{N}_7(\mathbf{r}_{e8})$ is not zero either since $\mathbf{W}_1$ is not parallel to edge 1 at element center. In view of the aforementioned problem, we propose to keep one basis ($\mathbf{N}_7$) the same as before, but modify the second basis ($\mathbf{N}_8$) as the following:

$$\begin{aligned} \hat{e}_7 &= \vec{v}_{23}/\|\vec{v}_{23}\|, \quad \mathbf{N}_7 = 4.5\xi_1\mathbf{W}_1 \\ \hat{e}_8 &= (\hat{z} \times \mathbf{W}_1)/\|\hat{z} \times \mathbf{W}_1\|, \quad \mathbf{N}_8 = c_8\xi_2\xi_3\nabla\xi_1, \end{aligned} \tag{2.20}$$

where $c_8$ is the normalization coefficient that makes $\hat{e}_8 \cdot \mathbf{N}_8(\mathbf{r}_{e8}) = 1$. In (2.20), instead of using the $\nabla\xi_1$ direction as $\hat{e}_8$, we employ the direction of $(\hat{z} \times \mathbf{W}_1)$. By doing so, $\hat{e}_8 \cdot \mathbf{N}_7(\mathbf{r}_{e8})$ is ensured to be zero. Furthermore, $\hat{e}_7 \cdot \mathbf{N}_8(\mathbf{r}_{e7}) = 0$ still holds true. In addition, with the choice of (2.20), the property of $\hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei}) = 0$ with $(i = 1, 2, ..., 6)$ and $(j = 7, 8)$ is still satisfied. Meanwhile, the property of $\hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei}) = 0$ with $(i = 7, 8)$ and $(j = 1, 2, ..., 6)$ is also satisfied since all the six edge vector bases vanish at the element center.

In summary, the six vector basis functions shown in (2.16) and the two vector bases given by (2.20) make a complete set of the first-order vector basis functions for a triangular element. Together with the unit vectors $\hat{e}_i$ defined in (2.16) and (2.20), they meet the requirements of (2.15), and hence making each entry in $\{e\}$ nothing but $\mathbf{E} \cdot \hat{e}_i(\mathbf{r}_{ei})$. It is also worth mentioning that the approach shown in (2.20) for modifying bases is equally applicable to other higher-order bases to make the unknown coefficient vector of the basis functions equal to the unknown electric field vector shown in (2.6).

### 2.3.3 Choice of H-points and H-directions

With the points and directions of the $\mathbf{E}$'s degrees of freedom known from the above section, it also becomes clear at which points and along which directions we evaluate

**H**. As shown in Fig. 2.2(a), for each $\hat{e}_i$ located at $\mathbf{r}_{ei}$, we draw a line perpendicular to $\hat{e}_i$ at $\mathbf{r}_{ei}$. On this line, we find two points such that the center point of the two points is $\mathbf{r}_{ei}$. The two points are where we need to prepare for **H** such that $\mathbf{E} \cdot \hat{e}_i$ can be accurately evaluated at $\mathbf{r}_{ei}$. For $\hat{e}_i$ located at the edge, the two points straddle the edge, and reside respectively in the two elements sharing the edge; for the internal degree of freedom whose $\hat{e}_i$ is located at the element center, both **H**-points are chosen inside the element. The union of the two points we find for each $\hat{e}_i$ makes the whole set of $\mathbf{r}_{hi}$ $(i = 1, 2, ..., N_h)$. As for the direction used at each **H**-point, for analyzing 2-D problems, it is $\hat{h}_i = \hat{z}$ $(i = 1, 2, ..., N_h)$.

Fig. 2.4 illustrates the locations of the **H**-points drawn for the **E** unknowns located in a single element. Basically, each **E** unknown is associated with a pair of **H**-points. Each pair is marked by a different color in Fig. 2.4. Coincident H-field points are permitted in the proposed algorithm. No extra checking to avoid overlapping points is needed.

The total number of **E** unknowns, i.e. the length of $\{e\}$ vector in (2.5), is $N_e = 2N_{edge} + 2N_{patch}$; whereas the total number of **H** unknowns, i.e. the length of $\{h\}$ vector, is $N_h = 10N_{patch}$ since there are 10 **H**-points in each patch.

### 2.3.4 Formulations of $\mathbf{S}_e$ and $\mathbf{S}_h$

$\mathbf{S}_e$ is a sparse matrix of size $N_h \times N_e$, whose $ij$-th entry can be written as

$$\mathbf{S}_{e,ij} = \hat{h}_i \cdot \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}), \tag{2.21}$$

where $i$ denotes the global index of the **H**-point, while $j$ is the global index of the **E**'s vector basis function. The number of nonzero elements in each row of $\mathbf{S}_e$ is 8 since the $H_z$ at each specified point is evaluated from the curl of **E** expanded into eight vector basis functions in the element where the **H**-point resides. When $\mathbf{S}_e$ is constructed, the elements share the same tangential **E**, i.e. $\{e\}$, in common along the edges, thus the tangential continuity of **E** is enforced during the construction of $\mathbf{S}_e$.

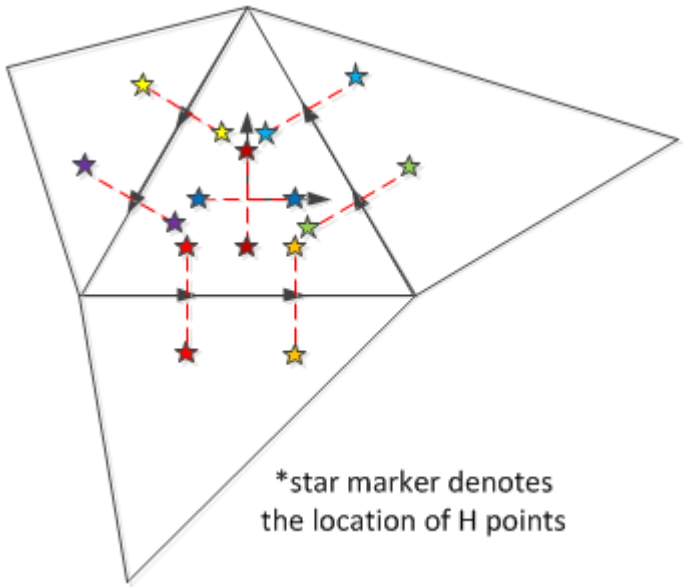



Fig. 2.4. Illustration of **H**-points (stars) for all **E**'s degrees of freedom (arrows) in one element.

The curl of each vector basis $\mathbf{N}_j$ in (2.21) can be evaluated analytically based on their expressions given in (2.16) and (2.20), and then the point $\mathbf{r}_{hi}$ is substituted into the resulting analytical expression to obtain the curl at the point.

The size of $\mathbf{S}_h$ is still the same as that of the transpose of $\mathbf{S}_e$, namely $N_e \times N_h$. However, it is not the transpose of $\mathbf{S}_e$. Consider an arbitrary **E**-unknown $e_i$, and denote the two **H**-unknowns associated with it to be $h_m$, and $h_n$ respectively. Assume the distance between $h_m$ and $h_n$ is $l_i$. Since the two **H**-points of each $e_i$ are positioned in a way as that shown in Fig. 2.4, the discretization of $\nabla \times \mathbf{H}$ for $e_i$ becomes $\pm(h_m - h_n)/l_i$. Therefore, every row of $\mathbf{S}_h$ has only two nonzero elements, whose entries are

$$\mathbf{S}_{h,ij} = \pm\frac{1}{l_i}, \tag{2.22}$$

where $j$ denotes the global index of the **H**-point associated with the $e_i$.

### 2.3.5 Time Marching Scheme and Stability Analysis

For a general unstructured mesh, if we choose $\mathbf{S}_h = \mathbf{S}_e^T$, the accuracy cannot be ensured. For an accurate $\mathbf{S}_h$ constructed in the proposed work, it is not the transpose of $\mathbf{S}_e$. The resultant $\mathbf{S}$ is not symmetric. As a result, the explicit marching like (2.11) and (2.12) or a central-difference based explicit time marching of (2.13) is absolutely unstable.

To understand the stability problem more clearly, we can perform a stability analysis of the central-difference based time discretization of (2.13) based on the approach given in [35, 44]. We start with a general inhomogeneous lossless problem since the analysis of a lossy problem can be done in a similar way. The $z$-transform of the central-difference based time marching of (2.13) results in the following equation:

$$(z-1)^2 + \Delta t^2 \lambda z = 0, \tag{2.23}$$

where $\lambda$ is the eigenvalue of $\mathbf{S}$. The two roots of (2.23) can be readily found as

$$z_{1,2} = \frac{2 - \Delta t^2 \lambda \pm \sqrt{\Delta t^2 \lambda (\Delta t^2 \lambda - 4)}}{2}. \tag{2.24}$$

If $\mathbf{S}$ is Hermitian positive semidefinite, its $\lambda$ is real and no less than zero. Thus, we can always find a time step to make $z$ in (2.24) bounded by 1, and hence the explicit simulation of (2.13) stable. Such a time step satisfies $\Delta t \leq 2/\sqrt{\lambda_{max}}$, where $\lambda_{max}$ is the maximum eigenvalue, which is also $\mathbf{S}$'s spectral radius. However, if $\mathbf{S}$ is not Hermitian positive semidefinite, its eigenvalues either are real or come in complex-conjugate pairs [45]. For complex-valued or negative eigenvalues $\lambda$, the two roots $z_1$ and $z_2$ shown in (2.24) 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 (2.13) must be unstable. Similarly, we can perform a stability analysis of a general lossy problem, and find the same conclusion—if the $\mathbf{S}$ is not symmetric and supports complex-valued and/or negative eigenvalues, the central-difference-based explicit timed-domain simulation of (2.13) is absolutely unstable.

The stability problem is solved in this work by developing a matrix-free time marching scheme that is stable. We will start with the following backward-difference-based discretization of (2.13) to explain the basic idea. But the final time marching equation only involves the field solutions at previous time steps for obtaining the field solution at current time step. The backward-difference-based discretization of of (2.13) results in

$$\{e\}^{n+1} - 2\{e\}^n + \{e\}^{n-1} + \Delta t diag(\{\frac{\sigma}{\epsilon}\})(\{e\}^{n+1} - \{e\}^n) + \Delta t^2 \mathbf{S}\{e\}^{n+1} = -\Delta t^2 diag(\{\frac{1}{\epsilon}\}) \left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}, \tag{2.25}$$

which is obtained by approximating both first- and second-order time derivatives by a backward-difference scheme [37]. Performing a stability analysis of (2.25), we find the two roots of $z$ as

$$z_{1,2} = \frac{1}{1 \pm j\Delta t\sqrt{\lambda}}. \tag{2.26}$$

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 $\lambda$ can be complex-valued or even negative. To make the magnitude of (2.26) bounded by 1, we find that the time step needs to satisfy the following condition

$$\Delta t > 2\frac{|\mathrm{Im}(\sqrt{\lambda})|}{|\sqrt{\lambda}|^2}, \tag{2.27}$$

where $\mathrm{Im}(\cdot)$ denotes the imaginary part of $(\cdot)$. Interestingly, the scheme is stable for large time step, but not stable for small time step. For real eigenvalues, it is absolutely stable. However, for complex or negative eigenvalues, to be stable, one should not choose a small time step that violates (2.27).

Rearranging the terms in (2.25), we obtain

$$\tilde{\mathbf{D}}\{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} + \Delta t diag(\{\frac{\sigma}{\epsilon}\})\{e\}^n - \Delta t^2 diag(\{\frac{1}{\epsilon}\}) \left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}, \tag{2.28}$$

where

$$\tilde{\mathbf{D}} = \mathbf{I} + \Delta t diag(\{\frac{\sigma}{\epsilon}\}) + \Delta t^2 \mathbf{S}. \tag{2.29}$$

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

$$\mathbf{D} = \mathbf{I} + \Delta t diag(\{\frac{\sigma}{\epsilon}\}). \tag{2.30}$$

Front multiplying both sides of (2.28) by $\mathbf{D}^{-1}$, we obtain

$$(\mathbf{I} + \tilde{\mathbf{M}})\{e\}^{n+1} = \mathbf{D}^{-1}\{f\}, \tag{2.31}$$

where $\{f\}$ is the right-hand side of (2.28), and

$$\tilde{\mathbf{M}} = \Delta t^2 \mathbf{D}^{-1}\mathbf{S}. \tag{2.32}$$

Although (2.28) permits the use of any large time step, we choose the time step in the following way

$$\Delta t^2 < \frac{1}{\|\mathbf{S}\|}, \tag{2.33}$$

and hence

$$\Delta t^2 \|\mathbf{S}\| < 1. \tag{2.34}$$

Notice that the time step determined from (2.33) is the same as that of a traditional explicit scheme for stability. This is also the time step required by accuracy when space step is determined based on the input spectrum. This is because the square root of spectral radius and thereby the norm of $\mathbf{S}$ corresponds to the largest frequency present in the system response. To capture this frequency accurately, a time step of (2.33) is necessary. It is also worth mentioning that the time step that violates (2.27) turns out to be very small in the proposed method since the imaginary part of the complex eigenvalues is negligible as compared to the real part, owing to the accuracy of the proposed space discretization scheme. Thus, (2.33) satisfies (2.27) in general.

The $\mathbf{D}$ is a diagonal matrix shown in (2.30). The norm of its inverse can be analytically evaluated as

$$\|\mathbf{D}^{-1}\| = 1/\min_{1\le i\le N_e}(1 + \Delta t\sigma_i/\epsilon_i) = 1. \tag{2.35}$$

we hence obtain, from (2.34) and (2.35),

$$\|\tilde{\mathbf{M}}\| = \Delta t^2 \|\mathbf{D}^{-1}\mathbf{S}\| \le \Delta t^2 \|\mathbf{D}^{-1}\| \|\mathbf{S}\| < 1. \tag{2.36}$$

As a result, we can evaluate the inverse of $\mathbf{I} + \tilde{\mathbf{M}}$ by

$$(\mathbf{I} + \tilde{\mathbf{M}})^{-1} = \mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \tilde{\mathbf{M}}^3 + \ldots, \tag{2.37}$$

which can be truncated since (2.36) is satisfied. Together with the fact that the mass matrix $\mathbf{D}$ is diagonal, and thus $\tilde{\mathbf{M}}$ does not involve any matrix inversion, the system matrix has an explicit inverse, and hence no matrix solutions are required in the proposed method. This is very different from an iterative matrix solution that does not have an explicit inverse of the system matrix. The (2.31) can then be computed as

$$\{e\}^{n+1} = (\mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \cdots + (-\tilde{\mathbf{M}})^k)\mathbf{D}_i\{f\}, \tag{2.38}$$

where $\mathbf{D}_i$ is diagonal matrix $\mathbf{D}$'s inverse. The number of terms $k$ is ensured to be small (less than 10) since (2.36) holds true. When mesh changes, the spectral radius of $\mathbf{S}$ changes. However, the time step required by accuracy or by a traditional explicit scheme for stability also changes. Since such a time step is chosen based on the criterion of (33), the convergence of (2.37) is guaranteed and the convergence rate does not depend on the mesh quality.

The computational cost of (2.38) is $k$ sparse matrix-vector multiplications since each term can be computed from the previous term. For example, after $\mathbf{D}_i\{f\}$ is computed, let the resultant be vector $y$, the second term in (2.38) 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 (2.38) is the cost of multiplying $-\tilde{\mathbf{M}}$ by the vector obtained at the previous step, thus the overall computational complexity is strictly linear (optimal).

When the proposed method is applied to a regular orthogonal grid, we do not need to add a few more sparse matrix-vector multiplications shown in (2.38). One sparse matrix-vector multiplication based on $\tilde{\mathbf{M}}$ is sufficient for stability. Only for unstructured meshes where complex-valued or negative eigenvalues exist, (2.38) is necessary for stability. The key for (2.38) to be free of matrix solution is the diagonal mass matrix created by the proposed new method for discretizing Maxwell's equations

in unstructured meshes. The same series expansion can be applied to the backward-difference-based TDFEM, but the resultant scheme still involves a matrix solution.

### 2.3.6 Imposing Boundary Conditions

The implementation of boundary conditions in the proposed method is similar to that in the TDFEM and FDTD, since the proposed method has a numerical system conformal to the two methods.

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 (2.5), 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 (2.9), since the $\{e\}$ entries at the $N_b$ points are known, the updating of (2.9) 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 (2.13), is the same as keeping just $(N_e - N_b)$ rows of **S**, providing the full-length $\{e\}$ (with the boundary entries specified) for the $\{e\}$ multiplied by **S**, but taking only the $N_e - N_b$ rows of all the other terms involved in (2.13). To impose a PMC to truncate the computational domain, the total **E** unknown number is $N_e$ without any reduction. The (2.5) is formulated as it is since the **H**-points having the PMC boundary condition can be placed outside the computational domain, instead of right on the boundary where **E** is located. As for (2.9), there is no need to make any change either since the tangential **H** is set to be zero outside the computational domain. For open-region problems, the framework of (2.5) and (2.9) 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.

## 2.4 Numerical Results

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

### 2.4.1 Wave Propagation in a 2-D Ring Mesh

A 2-D ring centered at (1.0 m, 1.0 m) with inner radius 0.5 m and outer radius 1.0 m is simulated in free space. The triangular mesh is generated by DistMesh [46], the details of which are shown in Fig. 2.5. The discretization results in 826 edges and 519 triangular patches. 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. Although the structure is irregular, we can use it to study a free-space wave propagation problem whose analytical solution is known. To do so, we impose an analytical boundary condition, i.e. the known value of tangential $\mathbf{E}$, on the boundary of the problem, which comprises the innermost and outermost circles; we then numerically simulate the fields inside the computational domain and correlate the results with the analytical solution.

The incident $\mathbf{E}$, which is also the total field in the given problem, is specified as $\mathbf{E} = \hat{y} f(t - x/c)$, where $f(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, $\tau = 2.5 \times 10^{-8}$ s, $t_0 = 4\tau$, and $c$ denotes the speed of light. The time step used in the proposed method is $\Delta t = 2.0 \times 10^{-11}$ s, which is the same as what a traditional central-difference based TDFEM has to use for stability. With this time step, the spectral radius of $\Delta t^2 \mathbf{S}$

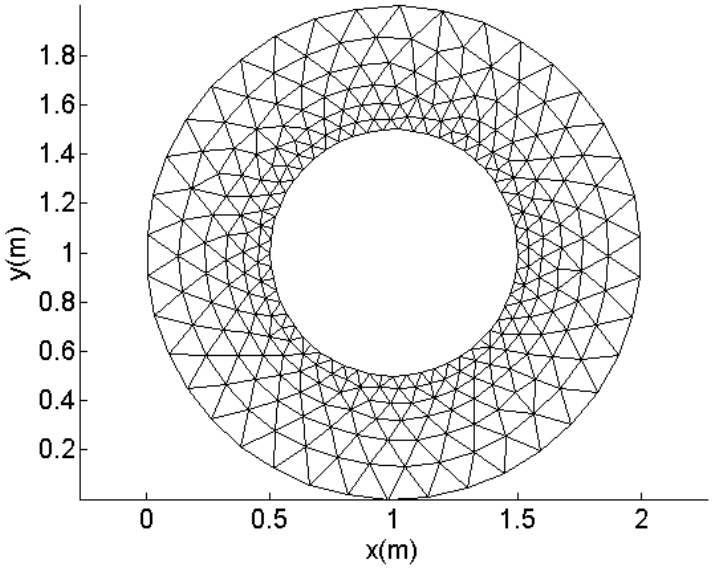



Fig. 2.5. Illustration of the mesh of a ring structure.

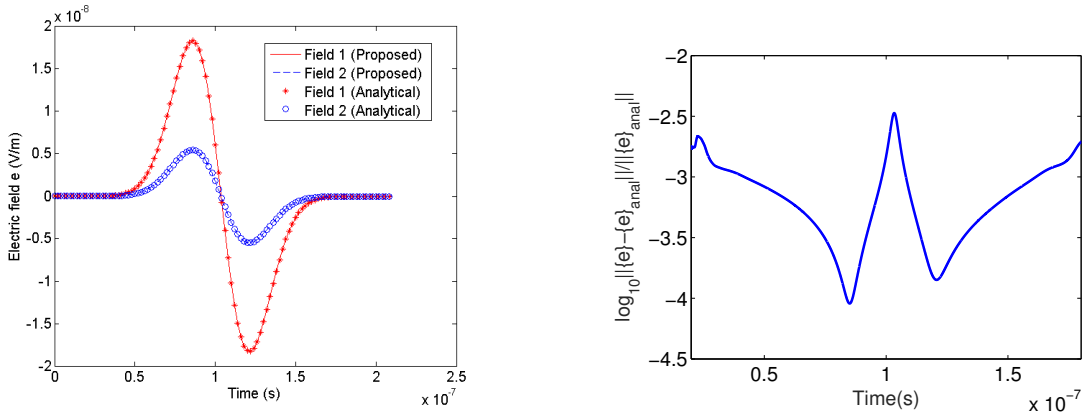
is 0.7359, and the number of expansion terms is 9 in (2.37). In Fig. 2.6(a), we plot the 2689- and 2690-th entry randomly selected from the unknown $\{e\}$ vector, which represent $\mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$, with $i = 2689$, and 2690 respectively. The point $\mathbf{r}_{ei}$ for both $i$ is (1.0789 m, 0.3497 m), thus the two $\mathbf{E}$ fields are sampled at the center point of one patch. From Fig. 2.6(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, we consider the relative error of the whole solution vector defined by

$$\text{Error}_{entire}(t) = \frac{||\{e\}_{this}(t) - \{e\}_{ref}(t)||}{||\{e\}_{ref}(t)||} \tag{2.39}$$

as a function of time, where $\{e\}_{this}(t)$ denotes the entire unknown vector $\{e\}$ of length $N_e$ solved from this method, while $\{e\}_{ref}(t)$ denotes the reference solution, which is analytical result $\{e\}_{anal}(t)$ in this example. The (2.39) allows us to evaluate the accuracy of the proposed method at all points for all time instants. In Fig. 2.6(b), we plot $\text{Error}_{entire}(t)$ across the whole time window in which the fields are not zero.

Notice that the vertical axis displays the error in $log_{10}$ scale, i.e. $log_{10}\text{Error}_{entire}(t)$. It is evident that less than 1% error is observed in the entire time window, demonstrating the accuracy of the proposed method. The center peak in Fig. 2.6(b) is due to the comparison with close to zero fields.




(a) (b)

Fig. 2.6. Simulation of ring mesh: (a) Electric fields simulated from the proposed method in comparison with analytical results. (b) $log_{10}$ of the entire solution error for all **E** unknowns v.s. time as compared to analytical result.

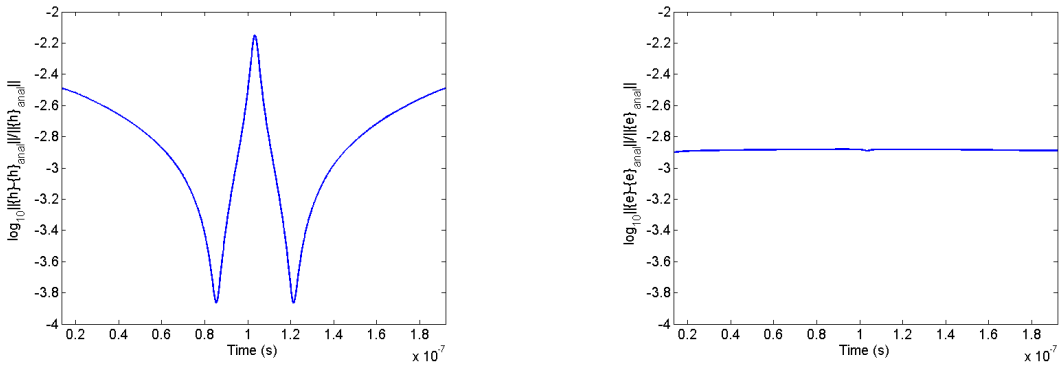
In addition to the accuracy of the entire method, we have also examined the accuracy of the individual $\mathbf{S}_e$, and $\mathbf{S}_h$ separately, since each is important to ensure the accuracy of the whole scheme. First, to solely assess the accuracy of $\mathbf{S}_e$, we perform the time marching of (2.5) only without (2.9) by providing an analytical $\{e\}$ to (2.5) at each time step. The resultant $\{h\}$ is then compared to analytical $\{h\}_{anal}$ at each time step. As can be seen from Fig. 2.7(a) where the following **H**-error

$$log_{10}\frac{||h(t) - h_{anal}(t)||}{||h_{anal}(t)||} \tag{2.40}$$

is plotted with respect to time, the error of all **H** unknowns is less than 1% 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 (2.9) only without (2.5) by providing an analytical $\{h\}$ to (2.9) at each time step. The

relative error of all **E** unknowns shown in (2.39) as compared to analytical solutions in $log10$ scale is plotted with time in Fig. 2.7(b). Again, less than 1% error is observed across the whole time window, verifying the accuracy of $\mathbf{S}_h$.




(a) (b)

Fig. 2.7. Simulation of ring mesh: (a) $log_{10}$ of the entire solution error v.s. time of all **H** unknowns obtained from $\mathbf{S}_e$-rows of equations. (b) $log_{10}$ of the entire solution error v.s. time of all **E** unknowns obtained from $\mathbf{S}_h$-rows of equations.

In this example, we have also varied the spacing between **H** points to examine its impact on time step and solution accuracy. Assume the $i$-th vector basis at point $\mathbf{r}_{ei}$ is shared by two elements $e1$ and $e2$. We draw a line passing $\mathbf{r}_{ei}$ and perpendicular to the edge where the vector basis resides. Assume the line intersects element $e1$ at point $\mathbf{r}_1$, and $e2$ at point $\mathbf{r}_2$. If $|\mathbf{r}_1 - \mathbf{r}_{ei}| < |\mathbf{r}_2 - \mathbf{r}_{ei}|$, then the distance between the two **H** points is set to be $(2|\mathbf{r}_1 - \mathbf{r}_{ei}|)/Hlratio$. With this definition, the smaller $Hlratio$, the larger the distance between the two **H** points, and the smallest $Hlratio$ one can choose is 1 for both points to fall inside the $e1$ and $e2$. As can be seen from Fig. 2.8, the solution accuracy is good irrespective of the choice of spacing, but larger spacing results in even better accuracy. This can be attributed to a less skewed discretization. The time step allowed by an explicit marching is $2.0 \times 10^{-11}$, $1.5 \times 10^{-11}$, and $10^{-11}$ s respectively for $Hlratio = 2$, 5, and 10. Hence, in general, a larger spacing is better for choice.

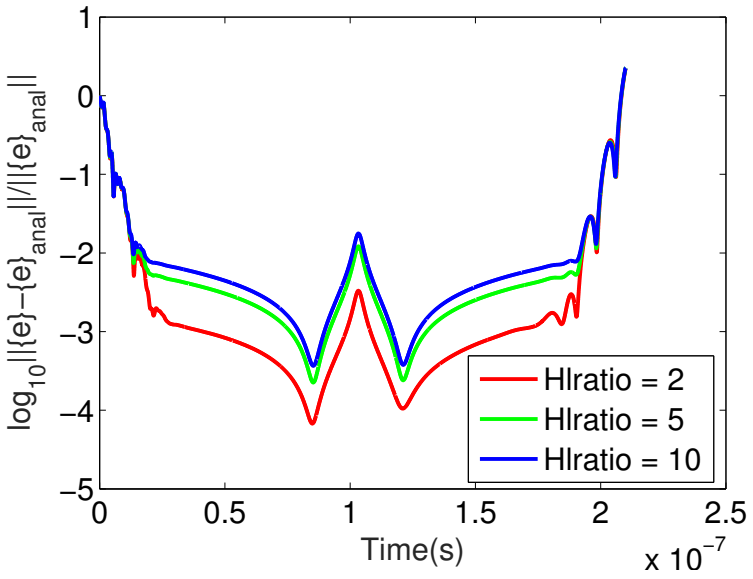



Fig. 2.8. $log_{10}$ of the entire solution error for all **E** unknowns v.s. time.

### 2.4.2 Wave Propagation in an Octagonal Spiral Inductor Mesh

The second example is a 1.5-turn octagonal spiral inductor in free space, whose 2-D mesh is shown in Fig. 2.9. The discretization results in 2081 edges and 1325 triangular patches. 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 **E** has the same form as that of the first example, but with $\tau = 2.0\times10^{-12}$ s in accordance with the new structure's dimension. The outermost boundary of the mesh is truncated by analytical **E** fields. The time step used is $\Delta t = 2.0 \times 10^{-16}$ s for simulating this $\mu$m-level structure, which is the same as that used in a traditional TDFEM method. This time step results in the spectral radius of $\Delta t^2\mathbf{S} = 0.8930$. The number of expansion terms is 9 in (2.37). The two degrees of freedom of the electric field located at one patch's center point, (206.83 $\mu$m, 12.65 $\mu$m), are plotted in Fig. 2.10(a) in comparison with analytical data. Excellent agreement can be observed.

In Fig. 2.10(b), we plot the entire solution error shown in (2.39) versus time, where the vertical axis displays the error in $log_{10}$ scale. Less than 3% error is observed in

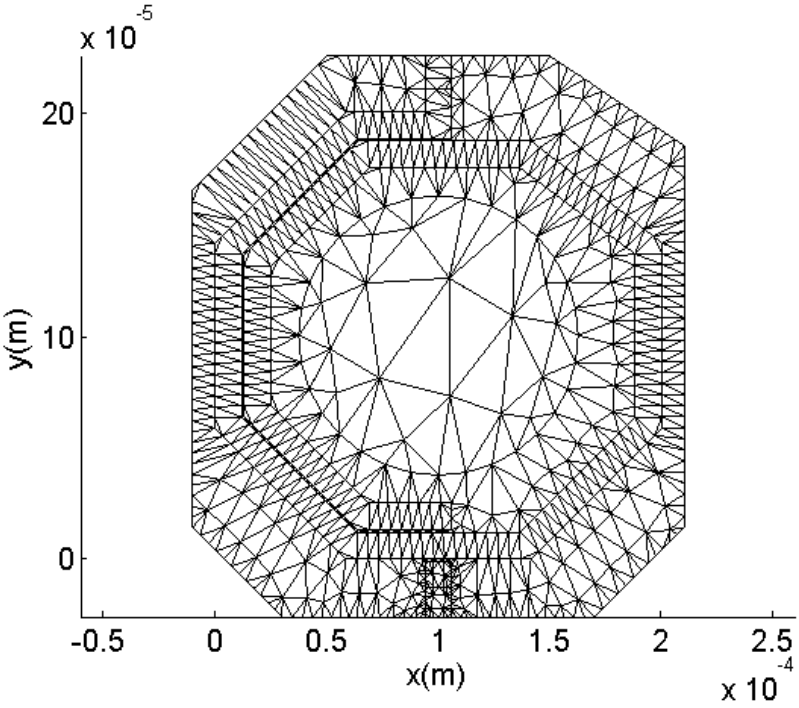
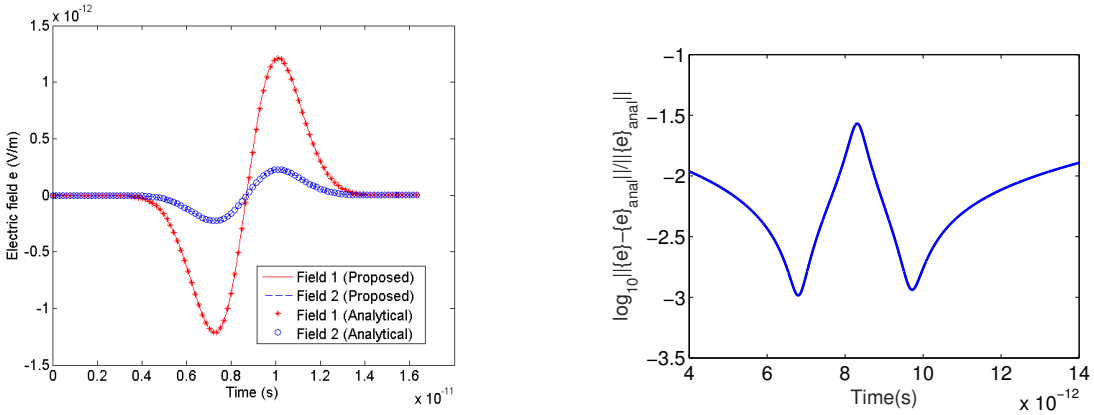



Fig. 2.9. Illustration of the mesh of an octagonal spiral inductor.




(a) (b)

Fig. 2.10. Simulation of an octagonal spiral inductor mesh: (a) Simulated electric field waveforms in comparison with analytical results. (b) $log_{10}$ of the entire solution error v.s. time as compared to analytical result.

the entire time window. It is evident that the proposed method is not just accurate at certain points, but accurate at all points in the computational domain for all time instants simulated. Note that the center peak error is due to zero passing, thus the comparison with close to zero fields at the specific time instant. The actual behavior at the zero-passing time instant is more objectively reflected in Fig. 2.10(a). In addition, we have examined the impact of $k$ on solution accuracy. We have enlarged $k$ from 9, to 18, and 36, the solution accuracy has no visible difference.

### 2.4.3 Wave Propagation and Reflection in an Inhomogeneous Medium

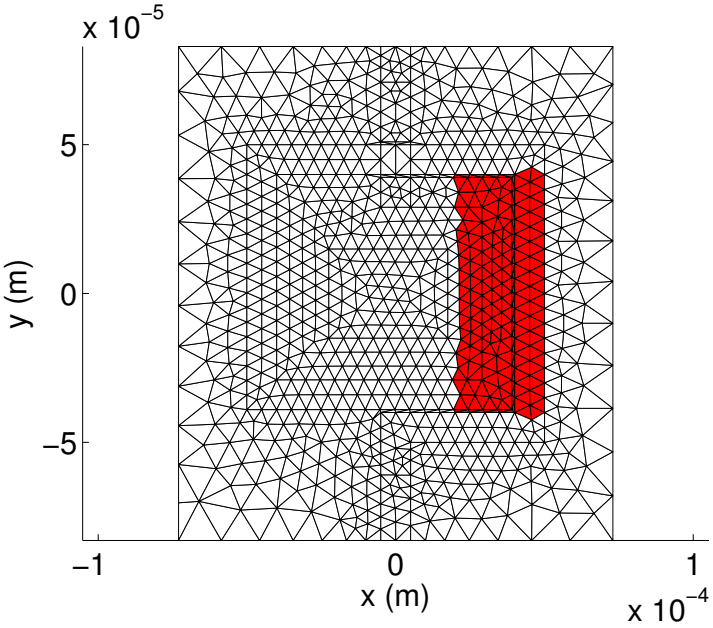
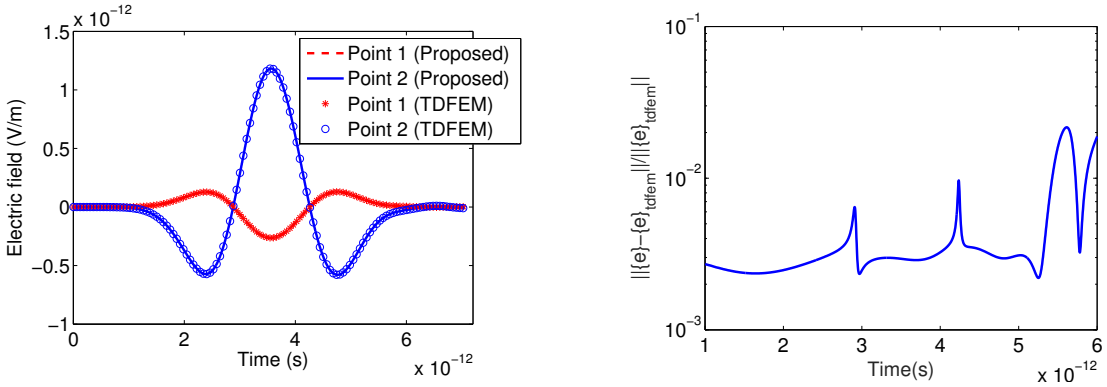



Fig. 2.11. Illustration of the mesh of a square inductor.

The third example is a wave propagation and reflection problem in an inductor mesh with dielectric materials. Fig. 2.11 displays the mesh details, where $\epsilon_r = 4$ in the red shaded region and 1 elsewhere. The top, bottom and right boundaries are terminated by perfect conductors, while the left boundary is truncated by the sum of the incident and reflected **E** fields. The incident **E** has the same form as that in the

first example, but with $\tau = 8.0\times10^{-13}$ s. The $\Delta t$ used is $5.0\times10^{-16}$ s, and the spectral radius of $\Delta t^2\mathbf{S}$ is 0.8119. The number of expansion terms is 9. In Fig. 2.12(a), the electric fields at two points $(-59.12, -71.31, 0)$ $\mu$m and $(-63.25, -64.3, 0)$ $\mu$m are plotted in comparison with TDFEM results. Excellent agreement can be observed. Again, such an agreement is also observed at all points for all time. As shown in Fig. 2.12(b), the entire solution error as compared with the TDFEM solution is less than 3% at all time instants even though the mesh is highly skewed. A few peak errors are due to the comparison with close-to-zero fields.




(a) (b)

Fig. 2.12. Simulation of a square inductor mesh: (a) Electric fields simulated from the proposed method in comparison with TDFEM results. (b) Entire solution error v.s. time as compared to reference TDFEM result.

### 2.4.4 Simulation of a PEC Cavity

The fourth example is a 2-D cavity. The cavity is filled with air and terminated by PEC on four sides. The mesh is shown in Fig. 2.13. We solve the transverse magnetic fields of TM11 mode for this cavity. The $\Delta t$ used is $2.0 \times 10^{-11}$ s. Nine terms are kept in (2.38). The same problem is also simulated using TDFEM for comparison. In Fig. 2.14(a), the magnetic field waveform at a randomly selected point (0.2415, 0.0145) m is plotted in comparison with analytical results. Excellent agreement can

be observed. Meanwhile, we calculate the entire solution error, which measures the error of the entire set of field unknowns, as compared with the analytical solution at each time step for both the proposed method and the TDFEM. The errors of the two methods are shown in Fig. 2.14(b) as a function of time. Obviously, both methods are accurate, and the proposed method is shown to have a better accuracy. This can be attributed to the better space discretization accuracy of the proposed method for the same mesh.

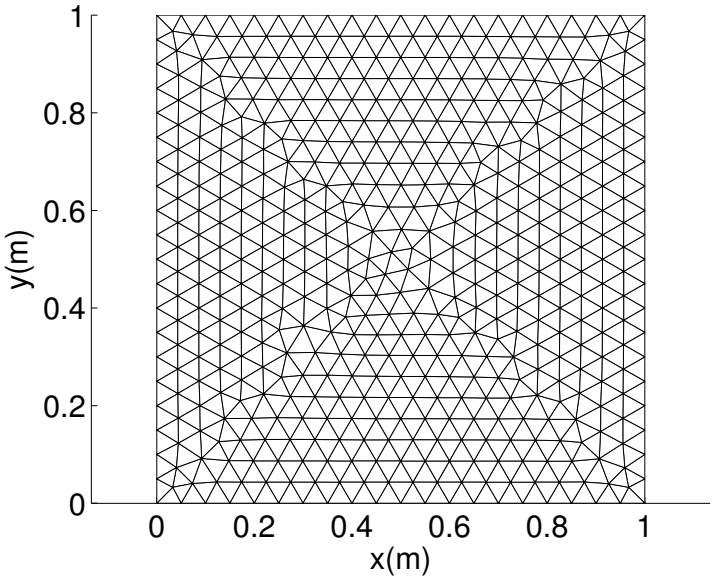
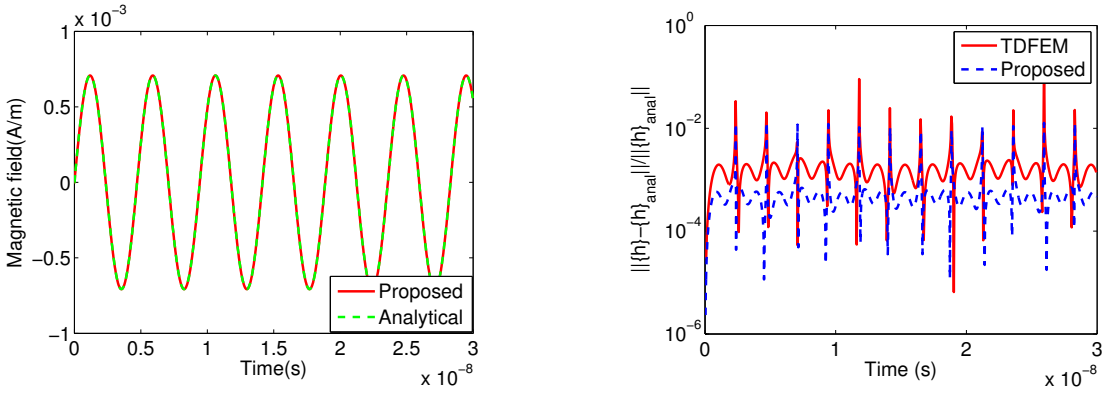



Fig. 2.13. Illustration of the mesh of a cavity.

### 2.4.5 Dependence of Error on Time Step Size

To analyze how the error depends on the time step size, we simulate a wave propagation problem in a 2-D circle, whose mesh is shown in Fig. 2.15(a). The incident **E** field has the same form as is shown in Section 2.4.1, but with $\tau = 2.0 \times 10^{-8}$ s. An explicit marching is stable for a time step no greater than $1.25 \times 10^{-11}$ s. Therefore, we choose the time step to be $1.25 \times 10^{-11}$ s, $6.25 \times 10^{-12}$ s, $3.125 \times 10^{-12}$ s respectively to run the simulation. In Fig. 2.15(b), the entire solution error



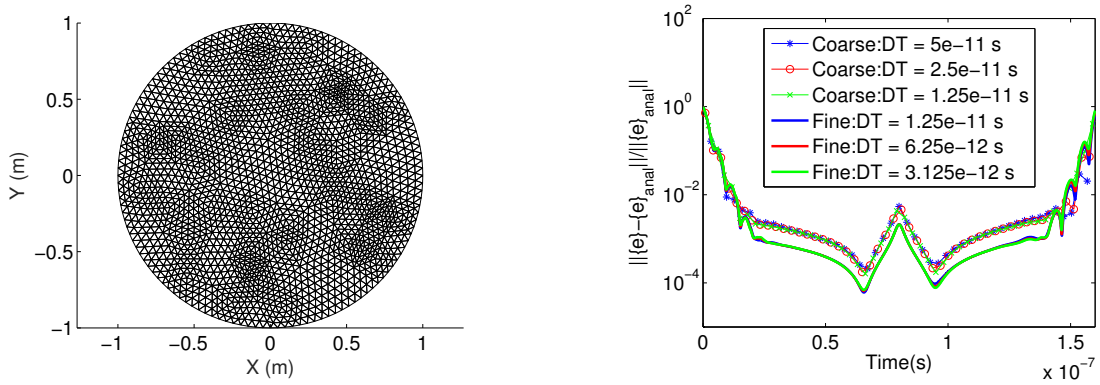

(a) (b)

Fig. 2.14. (a) Magnetic field of TM11 mode for a cavity simulated from the proposed method in comparison with analytical results. (b) Entire solution errors of the proposed method and the TDFEM v.s. time as compared to analytical results.

compared with analytical solution is plotted for different time step sizes. Obviously, the proposed method can produce accurate results for all three choices of time step. As the time step decreases, there is no significant improvement in accuracy since the time step allowed by a stable explicit marching is also the one required accuracy in the given mesh. However, the accuracy is improved more at time instants where the field solution has a more rapid temporal variation. This can be seen more clearly from the results generated from a coarser mesh, which are also plotted in Fig. 2.15(b).

### 2.4.6 Eigensolution of a Cavity Discretized into a Highly Unstructured Mesh

The previous examples are simulated for a certain excitation. One may be interested to know the accuracy for other excitations. The previous examples are all simulated in time domain. How about the accuracy in frequency domain? All these questions can be addressed by finding the eigenvalue solution of $\mathbf{S}$. This is because the field solution at any time and any frequency is a superposition of the eigenvectors




(a) (b)

Fig. 2.15. (a) Illustration of the fine mesh of a circle. (b) Entire solution errors v.s. time as compared to reference analytical results with the choice of different time steps for two meshes.

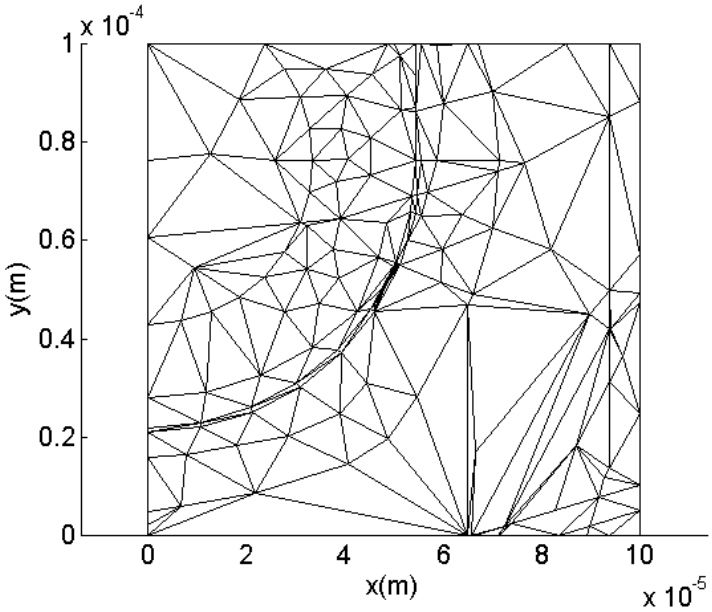



Fig. 2.16. Illustration of a highly irregular mesh.

of **S**, and the weight of each eigenvector can be determined from the corresponding eigenvalue. As a result, the correctness of the time-domain or frequency-domain results of the proposed method for any excitation can be found out by checking the

eigensolution of $\mathbf{S}$. We thus simulate a cavity whose analytical eigenvalues are known. The cavity is discretized into a highly irregular mesh as shown in Fig. 2.16 to examine the robustness of the proposed method in handling unstructured meshes. The mesh is provided by a semiconductor industry company from discretizing a real product. It appears to be of very poor quality because of accommodating all spatial features of the product, but is still a correct mesh.

We first construct matrices $\mathbf{S}_h$ and $\mathbf{S}_e$ separately, and then compute $\mathbf{S}$ based on (2.14), which is still a sparse matrix. We then find the eigensolution of $\mathbf{S}$ and compare the computed eigenvalues with analytical ones. The analytical eigenvalues can be found from the resonance frequencies of the cavity $\omega_r$ based on $\lambda = \omega_r^2$. In Table 2.1, the smallest 10 eigenvalues obtained from the proposed method are compared with analytical results in a descending order. It is clear that the proposed matrix-free method successfully generates accurate resonance frequencies despite the poor quality of the mesh. This example also serves as a good example to show that choosing $\mathbf{S}_h = \mathbf{S}_e^T$ would fail to produce accurate results in such an unstructured mesh, although the accuracy at some points for some excitations can be acceptable [47]. In the fourth and fifth column of Table 2.1, we list the eigenvalues computed by choosing $\mathbf{S}_h = \mathbf{S}_e^T$ and their relative errors as compared to analytical data. Comparing the last column with the third column, the effectiveness of the proposed method is obvious in obtaining good accuracy.

## 2.5 Conclusion

In this chapter, a new time-domain method having a naturally diagonal mass matrix is developed for solving Maxwell's equations. It is independent of element shape, thus suitable for analyzing arbitrarily shaped structures and materials discretized into unstructured meshes. The naturally diagonal mass matrix results in a strict linear computational complexity at each time step just like the complexity of an explicit FDTD method. Numerical experiments on various unstructured discretizations have

Table 2.1
Comparison of the smallest 10 eigenvalues of a cavity having a highly irregular mesh

| Analytical | This Method | Error (This) | $\mathbf{S}_h = \mathbf{S}_e^T$ | Error |
|---|---|---|---|---|
| 1.510e+27 | 1.451e+27 | 3.901e-02 | 1.064e+27 | 2.951e-01 |
| 1.421e+27 | 1.435e+27 | 9.909e-03 | 9.547e+26 | 3.282e-01 |
| 1.155e+27 | 1.178e+27 | 1.995e-02 | 7.516e+26 | 3.491e-01 |
| 8.883e+26 | 8.218e+26 | 7.482e-02 | 6.853e+26 | 2.285e-01 |
| 7.994e+26 | 8.180e+26 | 2.320e-02 | 6.134e+26 | 2.327e-01 |
| 7.106e+26 | 7.280e+26 | 2.454e-02 | 5.189e+26 | 2.697e-01 |
| 4.441e+26 | 4.372e+26 | 1.557e-02 | 3.296e+26 | 2.578e-01 |
| 3.553e+26 | 3.530e+26 | 6.457e-03 | 2.099e+26 | 4.090e-01 |
| 1.777e+26 | 1.806e+26 | 1.635e-02 | 9.152e+25 | 4.848e-01 |
| 8.883e+25 | 8.971e+25 | 9.913e-03 | 3.830e+25 | 5.688e-01 |

validated the accuracy and generality of the proposed method. This work has been successfully extended to 3-D analysis [48, 49], which will be presented in Chap. 3 and Chap. 4. It is also worth mentioning that the proposed method flexibly supports higher-order accuracy in both electric and magnetic fields. This can be achieved by using vector bases of any high order in each element to expand one field unknown, which consequently permits a higher-order discretization of the curl of the other field unknown in the loop area normal to the first field unknown.

# 3. MATRIX-FREE TIME-DOMAIN METHOD IN 3-D UNSTRUCTURED MESHES

## 3.1 Introduction

In Chap. 2, we develop a new matrix-free time-domain method, which requires no matrix solution, in unstructured meshes for general 2-D electromagnetic analysis. In this chapter, we extend it to perform electromagnetic analysis on 3-D structures. The method handles arbitrary unstructured meshes with the same ease as a finite-element method. Meanwhile, it is free of matrix solutions manifested by a naturally diagonal mass matrix, just like a finite-difference time-domain method. Modified vector bases for both tetrahedron and triangular prism are developed to directly connect the unknown coefficients of the vector basis functions employed to represent **E** (or **H**) with the unknowns obtained from the curl of **H** (or **E**), without any need for transformation. The proposed method employs only a single mesh. It does not require any interpolation and projection to obtain one field unknown from the other. Its accuracy and stability are guaranteed theoretically. Numerous experiments on unstructured triangular prism and tetrahedral meshes, involving both homogeneous and inhomogeneous and lossy materials, demonstrate the generality, accuracy, stability, and computational efficiency of the proposed method. The modified higher order vector bases developed in this chapter can also be used in any other method that employs higher order bases to obtain an explicit relationship between unknown fields and unknown coefficients of vector bases.

## 3.2 Proposed Method

Considering a general 3-D problem meshed into arbitrarily shaped elements, which can even be a mix of different shapes of element, we start from the differential form of Faraday's law and Ampere's law

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \tag{3.1}$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E} + \mathbf{J}, \tag{3.2}$$

we pursue a discretization of the two equations in time domain, such that the resultant numerical system is free of matrix solutions.

### 3.2.1 Discretization of Faraday's Law

In each element, we expand the electric field $\mathbf{E}$ in each element by vector bases $\mathbf{N}_j$ $(j = 1, 2, \ldots, m)$, as

$$\mathbf{E} = \sum_{j=1}^{m} u_j \mathbf{N}_j, \tag{3.3}$$

where $u_j$ is the $j$-th basis's unknown coefficient. Substituting (3.3) into (3.1) to evaluate $\mathbf{H}$ at $\mathbf{r}_{hi}$ point and along $\hat{h}_i$ direction, with $i = 1, 2, \ldots, N_h$, we obtain

$$\mathbf{S}_e\{u\} = -diag\left(\{\mu\}\right) \frac{\partial \{h\}}{\partial t}, \tag{3.4}$$

where $i$-th entry of vector $\{h\}$ is

$$h_i = \mathbf{H}(\mathbf{r}_{hi}) \cdot \hat{h}_i. \tag{3.5}$$

$\{u\}$ is of length $N_e$ consisting of all $u_j$ coefficients, $diag\left(\{\mu\}\right)$ is a diagonal matrix of permeability, and $\mathbf{S}_e$ is a sparse matrix having the following entry:

$$\mathbf{S}_{e,ij} = \hat{h}_i \cdot \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}). \tag{3.6}$$

Apparently, we have an infinite number of choices of $\mathbf{H}$ points and directions to build (3.4). However, to ensure the accuracy of the overall scheme which involves the

discretization of not only Faraday's law but also Ampere's law, we should select the **H** points and directions in such a way that the resultant **H** fields can, in turn, generate desired **E** accurately. Although there are many choices to do so, the simplest choice is to define a rectangular loop centering the **E** unknown and perpendicular to it, as shown in Fig. 3.1. Then, along this loop, we select the midpoint of each side as **H** point, and the unit vector tangential to each side as the **H**'s direction. The **H** fields obtained at these points and along these directions can certainly ensure the accuracy of **E** when we discretize Ampere's law. In addition, regardless of the element shape, there is no difficulty to define such a rectangular loop for each **E** unknown.

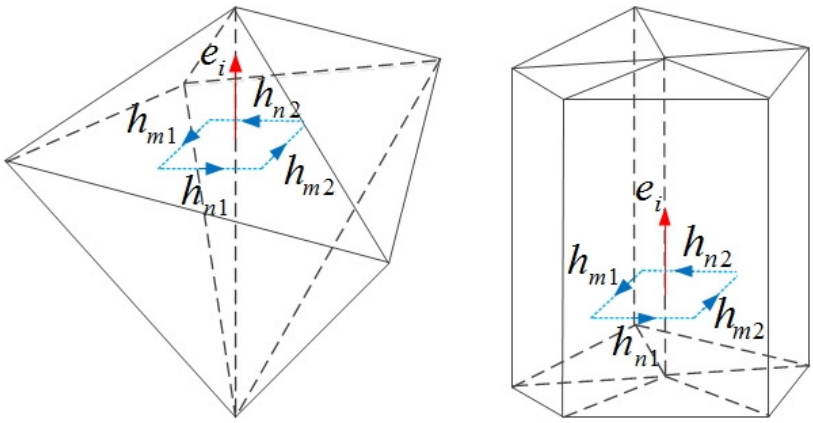



Fig. 3.1. Illustration of magnetic field points and directions for obtaining $e_i$.

### 3.2.2 Discretization of Ampere's Law

From Ampere's law, by evaluating **E** at $\mathbf{r}_{ei}$ point and along the $\hat{e}_i$ direction ($i = 1, 2, \ldots, N_e$), respectively, we obtain

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = \epsilon(\mathbf{r}_{ei})\frac{\partial e_i}{\partial t} + \sigma(\mathbf{r}_{ei})e_i + \hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei}), \tag{3.7}$$

in which

$$e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i, \tag{3.8}$$

Based on the choice of **H**-points and directions shown in Fig. 3.1, the $\hat{e}_i \cdot \nabla \times \mathbf{H}$ in (3.7) can be discretized accurately as

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = (h_{m1} + h_{m2})/l_{im} + (h_{n1} + h_{n2})/l_{in}, \tag{3.9}$$

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 shown in Fig. 3.1. With (3.9), (3.7) can be rewritten as

$$\mathbf{S}_h\{h\} = diag\left(\{\epsilon\}\right)\frac{\partial\{e\}}{\partial t} + diag\left(\{\sigma\}\right)\{e\} + \{j\}, \tag{3.10}$$

where $\{j\}$'s entries are $\hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei})$, and $diag\left(\{\epsilon\}\right)$ and $diag\left(\{\sigma\}\right)$ are the diagonal matrices whose entries are permittivity and conductivity, respectively. $\mathbf{S}_h$ is a sparse matrix of size $N_e \times N_h$, each row of which has four nonzero entries only being

$$\mathbf{S}_{h,ij} = 1/l_{ij}, \tag{3.11}$$

where $j$ is the global index of the **H** unknown used to generate $e_i$, and $l_{ij}$ is simply the distance between the **E** point ($\mathbf{r}_{ei}$) and the **H** point ($\mathbf{r}_{hj}$) multiplied by two.

### 3.2.3 Formulation of Modified Vector Basis Functions

Can we use zeroth-order vector basis functions in (3.3)? The answer is negative. This is because they produce a constant **H** field in each element. As a result, they fail to accurately generate the **H** fields at an arbitrary point along an arbitrary direction, and thereby at the points and along the directions desired for generating accurate **E**. For example, the **H** fields at the desired points along the desired directions shown in Fig. 3.1 cannot be accurately obtained from zeroth-order vector basis functions. Hence, we propose to use higher-order vector bases. However, they need modifications to satisfy

$$\{u\} = \{e\} \tag{3.12}$$

to connect (3.10) with (3.4) directly. As shown in (3.3), $\{u\}$ is the vector containing all the unknown coefficients of the vector basis functions; while $\{e\}$ is the vector

of discretized electric fields as shown in (3.8). They may not be the same. If we use normalized zeroth-order vector bases, $\{u\} = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$, and therefore, (3.12) is satisfied. However, higher-order curl-conforming bases [43] do not completely satisfy this property. In [50], we do not modify the original higher order vector bases. Instead, we find the relationship between $\{e\}$ and $\{u\}$, which is $\{e\} = \mathbf{P}\{u\}$, where $\mathbf{P}$ is a block diagonal matrix. We then use this relationship to connect (3.10) with (3.4). In [51], we show by developing a set of modified higher order vector basis, we can make $\{u\}$ equal to $\{e\}$, and hence bypassing the need for transformation. This saves the computational cost of generating the transformation matrix $\mathbf{P}$ and its related computation.

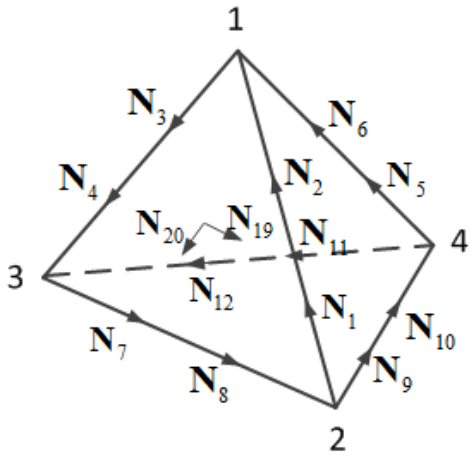



Fig. 3.2. Illustration of the degrees of freedom of the first-order curl-conforming vector bases in a tetrahedral element.

To see the point why higher-order curl-conforming bases do not satisfy (3.12) more clearly, we can substitute (3.3) into $e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$, obtaining

$$e_i = \sum_{j=1}^{m} u_j \mathbf{N}_j(\mathbf{r}_{ei}) \cdot \hat{e}_i. \tag{3.13}$$

Obviously, for (3.12) to be true, it is required that

$$\mathbf{N}_j(\mathbf{r}_{ei}) \cdot \hat{e}_i = \delta_{ji}. \tag{3.14}$$

In other words, the $j$-th vector basis's projection should be zero onto the direction and at the point associated with the $i$-th vector basis's degree of freedom. This property is naturally satisfied by edge vector basis functions. To explain, along any edge, the unit vector associated with the vector basis defined on this edge is tangential to the edge. Hence, (3.14) is naturally satisfied, since it is how the curl-conforming vector bases ensure the tangential continuity of the fields at the element interface. However, in higher-order vector bases, there also exist face vector basis functions and basis functions defined internal to the element. They, in general, do not satisfy the property of (3.14). Take the face vector bases as an example, their degrees of freedom are tangential to the face. However, each pair of the face vector bases is defined at the same point, and their directions are not perpendicular to each other. Hence, they do not satisfy the property of (3.14), and thus require modifications. Since first-order bases are sufficient for use in terms of generating second-order accuracy in the proposed method, next, we will use this set of bases as an example to show how to modify them. However, the essential idea applies to other higher-order bases.

In a tetrahedral element, there are 20 first-order vector bases [43]. Among them, 12 bases are edge vector basis functions, as shown in Fig. 3.2. They are defined as

$$\begin{aligned}
&\mathbf{N}_1 = (3\xi_2 - 1)\mathbf{W}_{21} \quad \mathbf{N}_2 = (3\xi_1 - 1)\mathbf{W}_{21} \\
&\mathbf{N}_3 = (3\xi_1 - 1)\mathbf{W}_{13} \quad \mathbf{N}_4 = (3\xi_3 - 1)\mathbf{W}_{13} \\
&\mathbf{N}_5 = (3\xi_4 - 1)\mathbf{W}_{41} \quad \mathbf{N}_6 = (3\xi_1 - 1)\mathbf{W}_{41} \\
&\mathbf{N}_7 = (3\xi_3 - 1)\mathbf{W}_{32} \quad \mathbf{N}_8 = (3\xi_2 - 1)\mathbf{W}_{32} \\
&\mathbf{N}_9 = (3\xi_2 - 1)\mathbf{W}_{24} \quad \mathbf{N}_{10} = (3\xi_4 - 1)\mathbf{W}_{24} \\
&\mathbf{N}_{11} = (3\xi_4 - 1)\mathbf{W}_{43} \quad \mathbf{N}_{12} = (3\xi_3 - 1)\mathbf{W}_{43}
\end{aligned} \tag{3.15}$$

where $\xi_i$ $(i = 1, 2, 3, 4)$ are volume coordinates at four vertices, and $\mathbf{W}_{ij}$ denotes the zeroth-order basis associated with the edge connecting vertex $i$ to vertex $j$.

Basically, along each edge, there are two degrees of freedom of the vector bases, located at the points $\mathbf{r}_{ei}$ whose distance is respectively 1/3, and 2/3 edge length to any one of the two nodes forming the edge. $\hat{e}_i$ associated with each edge basis is

simply the unit tangential vector of the edge where the basis is defined. The 12 edge bases satisfy the property of (3.14).

However, the other eight vector bases defined on the four faces of the tetrahedron do not satisfy the property of (3.14). These eight face bases can be written as

$$\begin{aligned} \mathbf{N}_{13} &= 4.5\xi_2\mathbf{W}_{43} \quad \mathbf{N}_{14} = 4.5\xi_3\mathbf{W}_{24} \\ \mathbf{N}_{15} &= 4.5\xi_3\mathbf{W}_{41} \quad \mathbf{N}_{16} = 4.5\xi_4\mathbf{W}_{13} \\ \mathbf{N}_{17} &= 4.5\xi_4\mathbf{W}_{21} \quad \mathbf{N}_{18} = 4.5\xi_1\mathbf{W}_{24} \\ \mathbf{N}_{19} &= 4.5\xi_1\mathbf{W}_{32} \quad \mathbf{N}_{20} = 4.5\xi_2\mathbf{W}_{13}. \end{aligned} \tag{3.16}$$

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

$$\begin{aligned} \hat{e}_{13} &= \hat{t}_{43} \quad \mathbf{r}_{13} = (\xi_2 = \xi_3 = \xi_4 = 1/3, \xi_1 = 0) \\ \hat{e}_{14} &= \hat{t}_{24} \quad \mathbf{r}_{14} = (\xi_2 = \xi_3 = \xi_4 = 1/3, \xi_1 = 0) \\ \hat{e}_{15} &= \hat{t}_{41} \quad \mathbf{r}_{15} = (\xi_1 = \xi_3 = \xi_4 = 1/3, \xi_2 = 0) \\ \hat{e}_{16} &= \hat{t}_{13} \quad \mathbf{r}_{16} = (\xi_1 = \xi_3 = \xi_4 = 1/3, \xi_2 = 0) \\ \hat{e}_{17} &= \hat{t}_{21} \quad \mathbf{r}_{17} = (\xi_1 = \xi_2 = \xi_4 = 1/3, \xi_3 = 0) \\ \hat{e}_{18} &= \hat{t}_{24} \quad \mathbf{r}_{18} = (\xi_1 = \xi_2 = \xi_4 = 1/3, \xi_3 = 0) \\ \hat{e}_{19} &= \hat{t}_{32} \quad \mathbf{r}_{19} = (\xi_1 = \xi_2 = \xi_3 = 1/3, \xi_4 = 0) \\ \hat{e}_{20} &= \hat{t}_{13} \quad \mathbf{r}_{20} = (\xi_1 = \xi_2 = \xi_3 = 1/3, \xi_4 = 0) \end{aligned} \tag{3.17}$$

in which $\hat{t}_{ij}$ stands for a unit tangential vector along the edge connecting vertex $i$ to vertex $j$. As can be seen, at the center of each face, there are two vector bases defined. Obviously, they do not satisfy the property of (3.14). For example, $\mathbf{N}_{19}(\mathbf{r}_{20}) \cdot \hat{e}_{20}$ is not zero. This is because at the center point of the face formed by nodes 1–3, $\mathbf{N}_{19}$ is not perpendicular to $\hat{e}_{20}$ whose direction is along the edge connecting vertex 1 to vertex 3.

If we rewrite (3.13) as

$$\{e\} = \mathbf{P}\{u\}. \tag{3.18}$$

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

$$\mathbf{P}_{ij} = \mathbf{N}_j(\mathbf{r}_{ei}) \cdot \hat{e}_i. \tag{3.19}$$

As shown in [50], with the first-order vector bases, $\mathbf{P}$ is a block diagonal matrix whose block size is either one or two. The diagonal block of size two corresponds to the two vector bases on each face, while each edge basis only corresponds to one diagonal entry, which is 1, in $\mathbf{P}$. Next, we show how to modify the face bases to make $\mathbf{P}$ an identity matrix.

Since the two face vector bases are defined at the same point, a linear combination of the two also makes a valid basis. The definitions of the face bases are hence not unique, which is also shown in [43]. We can modify them. To do so, we keep one face vector basis intact, but revise the other one. For a face having vertices $i$, $j$, and $k$, the two face bases we develop are

$$\mathbf{N}_{f_1} = 4.5\xi_i \mathbf{W}_{jk} \quad \hat{e}_{f_1} = \hat{t}_{jk} \tag{3.20}$$

$$\mathbf{N}_{f_2} = c\xi_j\xi_k\nabla\xi_i \quad \hat{e}_{f_2} = \frac{\hat{n}_f \times \mathbf{W}_{jk}}{||\hat{n}_f \times \mathbf{W}_{jk}||} \tag{3.21}$$

and for both face bases, their degrees of freedom are located at the face center, and hence

$$\mathbf{r}_{f_1} = \mathbf{r}_{f_2} = (\xi_i = \xi_j = \xi_k = 1/3). \tag{3.22}$$

Clearly, $\mathbf{N}_{f_1}$ in (3.20) is kept the same as before. It is the second face basis $\mathbf{N}_{f_2}$ that is changed. In (3.20), $\xi_i$ denotes the volume coordinate at node $i$, $\mathbf{W}_{jk}$ is the normalized zeroth-order edge basis with the superscripts denoting the two nodes of an edge, unit vector $\hat{t}_{jk}$ points from node $j$ to $k$. $c$ is the normalization coefficient making $\mathbf{N}_{f_2} \cdot \hat{e}_{f_2} = 1$ at the face center, and unit vector $\hat{n}_f$ is normal to the face.

With this modification, the revised first-order bases are equally complete, and meanwhile satisfying the desired property of (3.14). To see this point more clearly, now, we have

$$\begin{aligned} \mathbf{N}_{f_1}(\mathbf{r}_{f_2}) \cdot \hat{e}_{f_2} &= 0, \\ \mathbf{N}_{f_2}(\mathbf{r}_{f_1}) \cdot \hat{e}_{f_1} &= 0. \end{aligned} \tag{3.23}$$

The second row in the above holds true because $\nabla\xi_i$ is perpendicular to $\hat{t}_{jk}$. As a result, the original nonzero off-diagonal terms in $\mathbf{P}$ become zero. In addition to satisfying (3.23), we also have to ensure that the modified second face basis does not bring any new change to the original $\mathbf{P}$, i.e., changing the original zeros in $\mathbf{P}$ to nonzeros. If this happens, then the new bases defined in (3.20) cannot achieve the goal of making (3.12) true. This can be examined by evaluating the entries residing in the column and the row in $\mathbf{P}$ corresponding to the second new face basis, as other rows and columns are not affected. Essentially, we have to assess the following entries to see whether they are zero:

$$\begin{aligned} \mathbf{P}_{f_2,i} &= \mathbf{N}_{f_2}(\mathbf{r}_{ei}) \cdot \hat{e}_i, \quad (i \neq f_2) \\ \mathbf{P}_{i,f_2} &= \mathbf{N}_i(\mathbf{r}_{f_2}) \cdot \hat{e}_{f_2}. \quad (i \neq f_2) \end{aligned} \tag{3.24}$$

The entries of $\mathbf{P}_{f_2,i} = \mathbf{N}_{f_2}(\mathbf{r}_{ei}) \cdot \hat{e}_i$ reside on the row corresponding to the second face basis in $\mathbf{P}$. When the $\mathbf{r}_{ei}$ and $\hat{e}_i$ correspond to an edge basis, $\mathbf{N}_{f_2} = 0$ since $\xi_j\xi_k = 0$ on all edges except for the edge connecting $j$ to $k$. On this edge, $\mathbf{N}_{f_2}$ is perpendicular to the edge, and hence $\mathbf{N}_{f_2}(\mathbf{r}_{ei}) \cdot \hat{e}_i$ also vanishes. When $\mathbf{r}_{ei}$ and $\hat{e}_i$ belong to a face basis, $\mathbf{N}_{f_2} = 0$ since $\xi_j\xi_k = 0$ on all faces except for the two faces sharing edge connecting $j$ to $k$. On the same face where $\mathbf{N}_{f_2}$ is defined, as shown in (3.23), the corresponding $\mathbf{P}$ term is zero. On the other face, $\mathbf{N}_{f_2}$ is not zero, however, $\mathbf{N}_{f_2}$ is perpendicular to this face since it is along the direction of $\nabla\xi_i$. As a result, $\mathbf{N}_{f_2}(\mathbf{r}_{ei}) \cdot \hat{e}_i$ also vanishes. In summary, the modified new face basis preserves the original zeros in the row of this basis in $\mathbf{P}$, while vanishing the original nonzero entry in this row.

As for the entries of $\mathbf{P}_{i,f_2} = \mathbf{N}_i(\mathbf{r}_{f_2}) \cdot \hat{e}_{f_2}$, they are located in the column corresponding to the second face basis in $\mathbf{P}$. If basis $i$ is an edge basis, it is zero at the center points of three of the four faces and perpendicular to the fourth face. Hence, $\mathbf{P}_{i,f_2} = 0$. If basis $i$ is a face basis, it can be either the first face basis or the second face basis. If it is the first face basis, based on its expression shown in (3.20), among the other three faces where it is not located, it is zero on one of the three faces, and perpendicular to the rest two. Hence, $\mathbf{P}_{i,f_2} = 0$ if $i$-basis does not belong to the face

where $f_2$-basis is defined. If $i$-basis and $f_2$-basis belong to the same face, from (3.23), $\mathbf{P}_{i,f_2}$ is also zero. If basis $i$ is the second face basis, among the other three faces where it is not located, it is zero on two of the three faces, and perpendicular to the rest one. Hence, $\mathbf{P}_{i,f_2}$ is also zero. As a result, the new change of the second face basis also preserves the original zeros in the column corresponding to the second face basis in $\mathbf{P}$, while vanishing the original nonzero entry in this column.

Based on (3.20), the complete set of modified face bases and their projection directions, in accordance with the notations of (3.16), can be written as follows:

$$\begin{aligned}
\mathbf{N}_{14} &= c_{14}\xi_3\xi_4\nabla\xi_2 \quad \hat{e}_{14} = \frac{(\hat{n}_{234}\times\mathbf{W}_{43})}{||\hat{n}_{234}\times\mathbf{W}_{43}||} \\
\mathbf{N}_{16} &= c_{16}\xi_1\xi_4\nabla\xi_3 \quad \hat{e}_{16} = \frac{(\hat{n}_{134}\times\mathbf{W}_{41})}{||\hat{n}_{134}\times\mathbf{W}_{41}||} \\
\mathbf{N}_{18} &= c_{18}\xi_1\xi_2\nabla\xi_4 \quad \hat{e}_{18} = \frac{(\hat{n}_{124}\times\mathbf{W}_{21})}{||\hat{n}_{124}\times\mathbf{W}_{21}||} \\
\mathbf{N}_{20} &= c_{20}\xi_2\xi_3\nabla\xi_1 \quad \hat{e}_{20} = \frac{(\hat{n}_{123}\times\mathbf{W}_{32})}{||\hat{n}_{123}\times\mathbf{W}_{32}||}
\end{aligned} \tag{3.25}$$

where $\hat{n}_{ijk}$ denotes a unit vector normal to the face formed by vertices $i$, $j$, and $k$.

The basic idea of the aforementioned approach to make $\hat{e}_i \cdot \mathbf{N}_j(\mathbf{r}_{ei}) = \delta_{ij}$ satisfied is to choose appropriate basis direction and projection direction of the second basis, when encountering a pair of bases defined at the same point. The projection direction of the second basis is chosen perpendicular to the first basis at the point where the second basis's degree of the freedom is located. Meanwhile, the basis direction of the second basis is chosen to be perpendicular to the projection direction of the first basis. The essential idea of this approach is equally applicable to higher-order bases in other types of elements such as the triangular prism elements.

In a triangular prism element, there are 36 first-order vector bases. Among them, the three pairs of degrees of freedom located at the center of the top triangular face, the prism center, and the center of the bottom triangular face do not satisfy (3.14), while other bases satisfy. Similar to the treatment in a tetrahedron element, for the three sets, we keep the first basis, but modify the second basis. Take the top face

formed by nodes 1–3 as an example, we construct the following two bases and their projection directions:

$$\mathbf{N}_{f_1} = 4.5\xi_1\zeta_1(2\zeta_1 - 1)\mathbf{W}_{23}; \hat{e}_{f_1} = \hat{t}_{23} \tag{3.26}$$

$$\mathbf{N}_{f_2} = c\xi_2\xi_3\zeta_1(2\zeta_1 - 1)\nabla\xi_1; \hat{e}_{f_2} = \frac{(\hat{n}_f \times \mathbf{W}_{23})}{||\hat{n}_f \times \mathbf{W}_{23}||}. \tag{3.27}$$

Here, $\zeta_1 = 1$ on the top triangle and 0 on the lower one, $\mathbf{W}_{23}$ is the normalized zeroth-order basis defined on the edge connecting node 2 to node 3.

With the vector bases developed in the above, the entries in sparse matrix $\mathbf{S}_e$ shown in (3.6) can be determined. Since each vector basis $\mathbf{N}_j$ has an analytical expression, the $\nabla \times \mathbf{N}_j$ and thereby $\mathbf{S}_e$ can be analytically evaluated. In addition, when building $\mathbf{S}_e$, the tangential continuity of the electric fields is rigorously enforced at the element interface, since $\{u\}$, which is also $\{e\}$ now with the newly developed modified bases, is shared in common by adjacent elements. This is the same as how an FEM ensures the tangential continuity of the electric field.

### 3.2.4 Matrix-Free Time Marching

With $\{u\} = \{e\}$, the (3.4) and (3.10) can be solved in a leapfrog way, which requires no matrix solutions. The two can also be combined to solve as the following:

$$\frac{\partial^2 \{e\}}{\partial t^2} + diag\left(\left\{\frac{\sigma}{\epsilon}\right\}\right)\frac{\partial \{e\}}{\partial t} + \mathbf{S}\{e\} = -diag\left(\left\{\frac{1}{\epsilon}\right\}\right)\frac{\partial\{j\}}{\partial t}, \tag{3.28}$$

where

$$\mathbf{S} = diag\left(\left\{\frac{1}{\epsilon}\right\}\right)\mathbf{S}_h diag\left(\left\{\frac{1}{\mu}\right\}\right)\mathbf{S}_e. \tag{3.29}$$

Obviously, the matrices in front of the second- and first-order time derivatives are both diagonal. Hence, the proposed method possesses a naturally diagonal mass matrix. Therefore, an explicit marching of (3.28), such as a central-difference-based time marching, is free of matrix solutions. However, a brute-force explicit marching of (3.28) is absolutely unstable, because $\mathbf{S}$ is not symmetric in an unstructured mesh and it can support complex-valued and even negative eigenvalues. This has been proved in [50].

The stability problem can be solved as follows. Basically, we can begin with the following backward-difference-based time marching of (3.28)

$$\{e\}^{n+1} - 2\{e\}^n + \{e\}^{n-1} + \Delta t diag\left(\left\{\frac{\sigma}{\epsilon}\right\}\right)\left(\{e\}^{n+1} - \{e\}^n\right) + \Delta t^2 \mathbf{S}\{e\}^{n+1} = -\Delta t^2 diag\left(\left\{\frac{1}{\epsilon}\right\}\right)\left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}. \tag{3.30}$$

Rearranging the terms in (3.30), we obtain

$$\left(\mathbf{D} + \Delta t^2 \mathbf{S}\right)\{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} + \Delta t diag\left(\left\{\frac{\sigma}{\epsilon}\right\}\right)\{e\}^n - \Delta t^2 diag\left(\left\{\frac{1}{\epsilon}\right\}\right)\left(\frac{\partial\{j\}}{\partial t}\right)^{n+1} \tag{3.31}$$

where

$$\mathbf{D} = \mathbf{I} + \Delta t diag\left(\left\{\frac{\sigma}{\epsilon}\right\}\right), \tag{3.32}$$

which is diagonal. Front multiplying both sides of (3.31) by $\mathbf{D}^{-1}$, we obtain

$$(\mathbf{I} + \tilde{\mathbf{M}})\{e\}^{n+1} = \mathbf{D}^{-1}\{f\}, \tag{3.33}$$

where

$$\tilde{\mathbf{M}} = \Delta t^2 \mathbf{D}^{-1}\mathbf{S}, \tag{3.34}$$

and $\{f\}$ is the right hand side of (3.31).

Although the backward-difference-based (3.31) is stable for an infinitely large time step as analyzed in [50], we choose a time step based on the stability criterion of traditional explicit time marching. This time step satisfies

$$\Delta t < \frac{1}{\sqrt{\rho(\mathbf{S})}}. \tag{3.35}$$

It is also the time step required by accuracy when there is no fine feature relative to working wavelength, since the maximum eigenvalue's square root, $\sqrt{|\lambda_{max}|}$, corresponds to the maximum angular frequency present in the system response. With such a choice of time step, the spectral radius of $\tilde{\mathbf{M}}$ is guaranteed to be less than 1. This is because in this case, time step satisfies (3.35), and hence

$$\Delta t^2 \rho\left(\mathbf{S}\right) < 1, \tag{3.36}$$

in which $\rho(\cdot)$ denotes the spectral radius, which is the modulus of the largest eigenvalue. $\mathbf{D}$ is a diagonal matrix shown in (3.32). Hence,

$$\rho\left(\mathbf{D}^{-1}\right) = \frac{1}{\min_{1\leq i\leq N_e}(1+\Delta t\sigma_i/\epsilon_i)} = 1. \tag{3.37}$$

We therefore obtain from (3.36) and (3.37)

$$\rho\left(\tilde{\mathbf{M}}\right) = \Delta t^2 \rho\left(\mathbf{D}^{-1}\mathbf{S}\right) \leq \Delta t^2 \rho\left(\mathbf{D}^{-1}\right)\rho\left(\mathbf{S}\right) < 1. \tag{3.38}$$

As a result, without loss of accuracy, the inverse of $\mathbf{I}+\tilde{\mathbf{M}}$ can be evaluated by

$$(\mathbf{I}+\tilde{\mathbf{M}})^{-1} = \mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \tilde{\mathbf{M}}^3 + \ldots + (-\tilde{\mathbf{M}})^k, \tag{3.39}$$

where $k$ is guaranteed to be small since (3.38) is satisfied. Thus, the system matrix has an explicit inverse, and hence no matrix solutions are required. Equation (3.33) can then be computed as

$$\{e\}^{n+1} = (\mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \cdots + (-\tilde{\mathbf{M}})^k)\mathbf{D}_i\{f\}, \tag{3.40}$$

where $\mathbf{D}_i$ is diagonal matrix $\mathbf{D}$'s inverse. The computational cost of (3.40) is $k$ sparse matrix-vector multiplications, since each term can be computed from the previous term recursively, thus efficient.

## 3.3 Numerical Results

To validate the proposed new formulation-based matrix-free method, in this section, we simulate a variety of 3-D unstructured meshes. The aspect ratio of the mesh is defined as the longest edge length divided by the shortest edge length. The number of expansion terms $k$ used in (3.39) is nine for all of the examples simulated. The time step chosen is the same as that of the central-difference-based TDFEM.

### 3.3.1 Wave Propagation in a Tetrahedral Mesh of a 3-D Box

The first example is a 3-D free-space box of dimension $1\times0.5\times0.75$ m$^3$ discretized into tetrahedral elements. Its mesh is shown in Fig. 3.3 with 350 tetrahedral elements

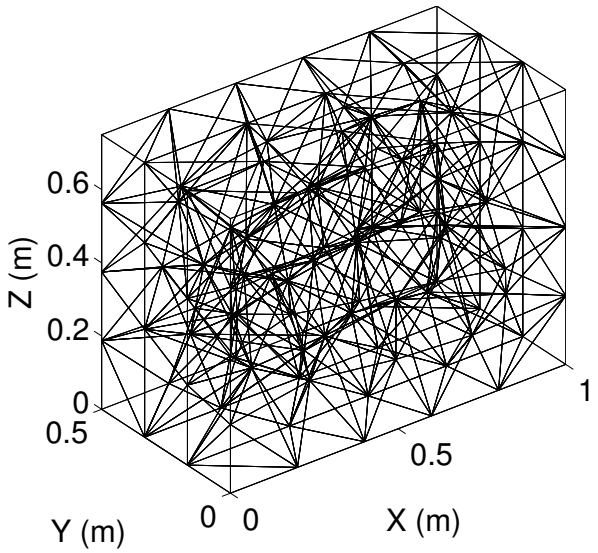
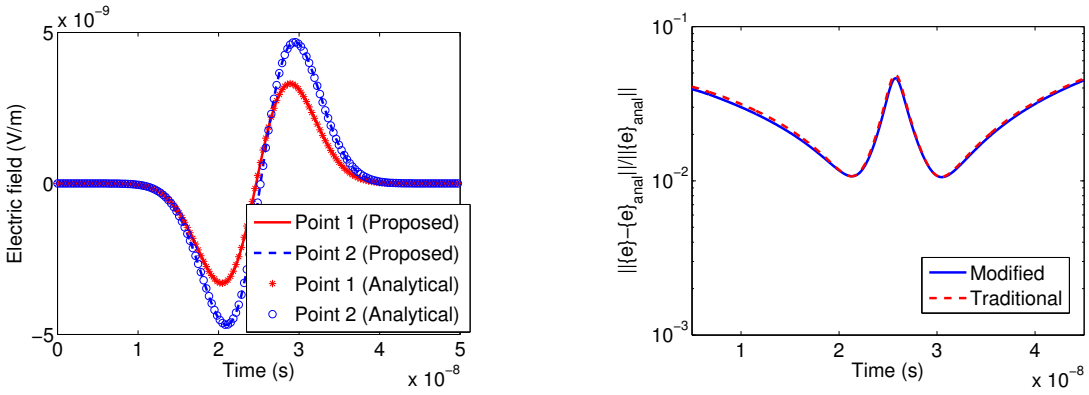



Fig. 3.3. Illustration of the tetrahedron mesh of a $1 \times 0.5 \times 0.75$ $\mathrm{m}^3$ rectangular box.

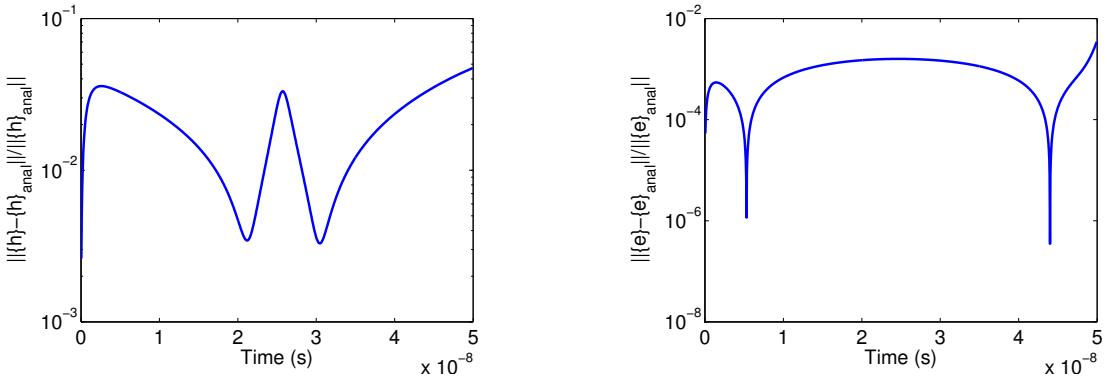
and 544 edges. The aspect ratio of the tetrahedral mesh is 3.67. To assess the accuracy of the proposed method, we simulate a free-space wave propagation problem, since its analytical solution is known. The incident $\mathbf{E}$, which is also the total field in the given




(a) (b)

Fig. 3.4. Simulation of a 3-D rectangular box discretized into tetrahedral elements: (a) Electric fields simulated from the proposed method as compared with analytical results. (b) Entire solution error as a function of time.

problem, is specified as $\mathbf{E} = \hat{y} f(t - x/c_0)$, where $f(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, $\tau = 6.0 \times 10^{-9}$ s, $t_0 = 4\tau$, and $c_0$ is the speed of light. The time step is chosen as $\Delta t = 1.6 \times 10^{-11}$ s. The proposed method takes only 2.12 MB to store sparse matrices $\mathbf{S}_e$ and $\mathbf{S}_h$, and $5.2 \times 10^{-4}$ s to finish the simulation at one time step. In Fig. 3.4(a), we plot the 1-st and 1,832-th entries randomly selected from the unknown $\{e\}$ vector, which represent $\mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$ with $i = 1$, and 1,832 respectively. It can be seen clearly that the electric fields solved from the proposed method agree very well with the analytical results.




(a) (b)

Fig. 3.5. (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 $\mathbf{E}$ unknowns obtained from $\mathbf{S}_h$-rows of equations.

To examine the accuracy of all unknowns solved from the proposed method, and also across all time instants, we consider the relative error of the whole solution vector defined by

$$\text{Error}_{entire}(t) = \frac{||\{e\}_{this}(t) - \{e\}_{ref}(t)||}{||\{e\}_{ref}(t)||} \tag{3.41}$$

as a function of time, where $\{e\}_{this}(t)$ denotes the entire unknown vector $\{e\}$ of length $N_e$ obtained from this method, whereas $\{e\}_{ref}(t)$ denotes the reference solution, which is analytical result $\{e\}_{anal}(t)$ in this example. In Fig. 3.4(b), we plot $\text{Error}_{entire}(t)$ across the 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. 3.4(b) is due to the comparison with close to zero fields.

This example has also been simulated in [50]. In Fig. 3.4(b), we compare the accuracy of the proposed new formulation with the formulation given in [50] . Obviously, the proposed new formulation with modified vector bases exhibits the same accuracy as the formulation given in [50].

In addition to the accuracy of the entire method, we have also examined the accuracy of the $\mathbf{S}_e$, and $\mathbf{S}_h$ individually, 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 (3.4) only without (3.10) by providing an analytical $\{e\}$ to (3.4) at each time step. The resultant $\{h\}$ is then compared to analytical $\{h\}_{anal}$ at each time step. As can be seen from Fig. 3.5(a), where the following entire $\mathbf{H}$ solution error

$$\frac{||h(t) - h_{anal}(t)||}{||h_{anal}(t)||} \tag{3.42}$$

is plotted with respect to time, the error of all $\mathbf{H}$ unknowns is $< 3\%$ 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 (3.10) only without (3.4) by providing an analytical $\{h\}$ to (3.10) at each time step. In Fig. 3.5(b), we plot (3.41) versus time. Again, very good accuracy is observed across the whole time window, verifying the accuracy of $\mathbf{S}_h$.

### 3.3.2 Wave Propagation in a Tetrahedral Mesh of a Sphere

The second example is a sphere of radius 0.24 m centering at the origin. It is discretized into tetrahedral elements in free space, whose 3-D mesh is shown in Fig. 3.6. The mesh consists of 1,987 tetrahedrons and 3,183 edges. The aspect ratio of the tetrahedral mesh is 6.19. The outermost boundary is truncated by analytically known electric fields. The time step is $\Delta t = 2.0 \times 10^{-12}$ s. The same incident $\mathbf{E}$ is as that in the first example is used, but $\tau = 2.0 \times 10^{-9}$ s is chosen in accordance with the new structure's dimension.

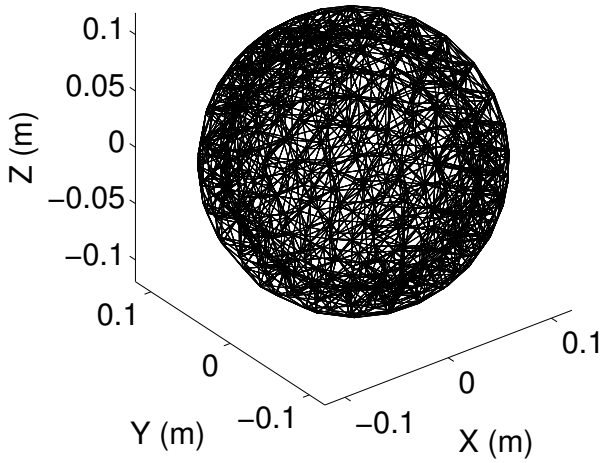
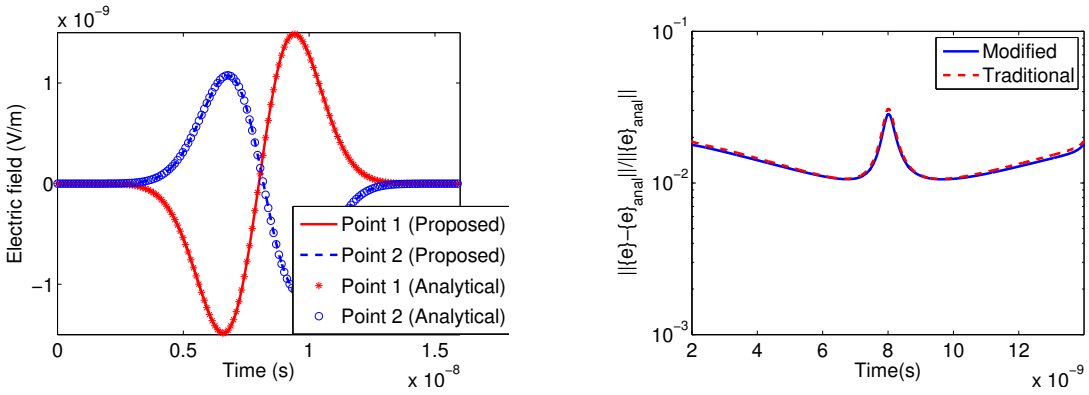



Fig. 3.6. Illustration of the tetrahedron mesh of a solid sphere.




(a) (b)

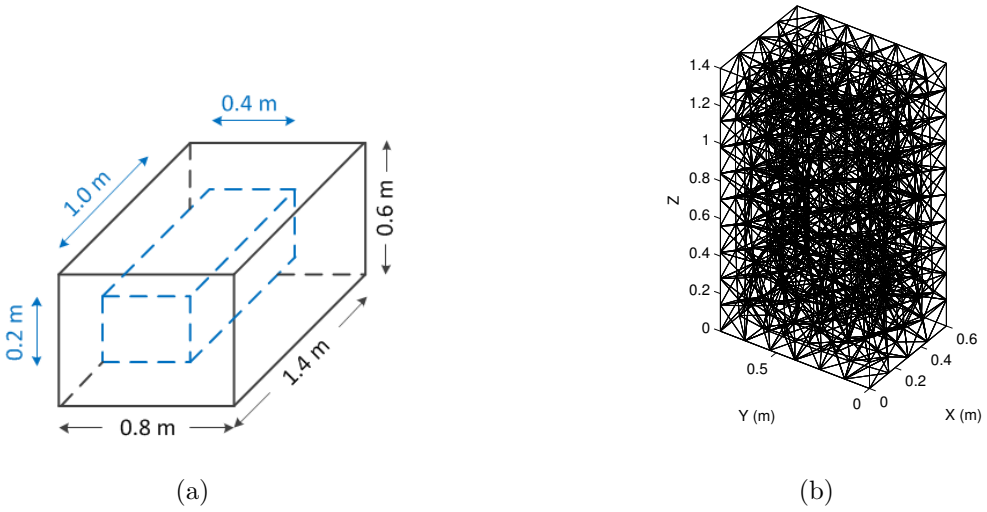
Fig. 3.7. Simulation of a sphere discretized into tetrahedral elements: (a) Electric fields obtaeind from the proposed method as compared with analytical results. (b) Entire solution error as a function of time for **E**.

The proposed method takes only 10.07 MB to store sparse matrices $\mathbf{S}_e$ and $\mathbf{S}_h$, and 0.003 s to finish the simulation at one time step. Two randomly selected electric

field unknowns, whose indices are 1 and 9,762 in $\{e\}$, are shown in Fig. 3.7(a) against analytical data. Excellent agreement can be seen.

In Fig. 3.7(b), the entire solution error shown in (3.41) is plotted as a function of time, which is shown to be less than 3%. To compare the accuracy of the proposed new formulation having modified vector bases with that of the traditional vector bases in [50], the entire solution error obtained by the formulation in [50] is also shown in Fig. 3.7(b). Obviously, the two exhibit the same accuracy, validating the proposed new vector bases, and its resulting matrix-free formulation.

### 3.3.3 Wave Propagation in a Tetrahedral Mesh of a Rectangular Box with a Hole

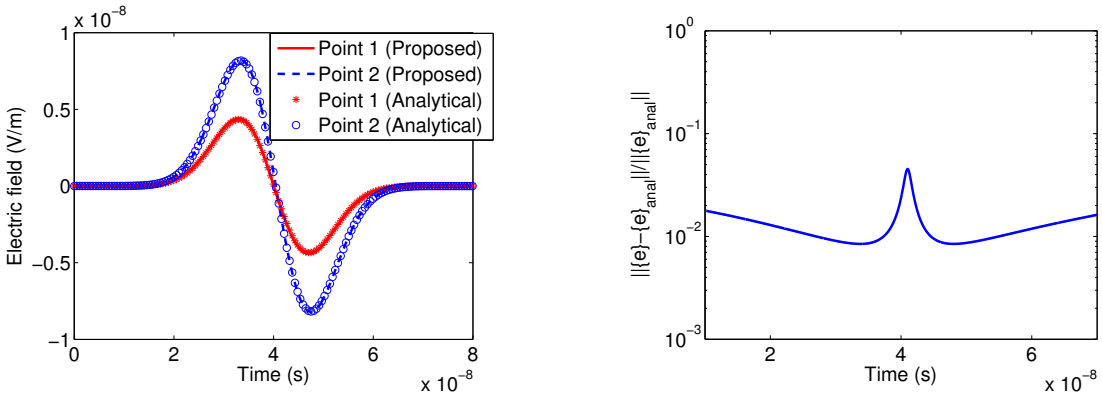



(a) (b)

Fig. 3.8. Illustration of a rectangular box with a hole: (a) Geometry. (b) Mesh Details.

The third example is a rectangular box whose size is $0.6 \times 0.8 \times 1.4$ m$^3$ with a hole in the center, whose structure is shown in Fig. 3.8(a). Its mesh is shown in Fig. 3.8(b). The shape of the hole is also a rectangular box but of size $0.2 \times 0.4 \times 1.0$ m. It is discretized into tetrahedral elements having 1,637 tetrahedrons and 2,456 edges. The

aspect ratio of the tetrahedral mesh is 5.36. The time step is chosen as $\Delta t = 2 \times 10^{-11}$ s. A free-space wave propagation problem is simulated in the given mesh, with the same incident $\mathbf{E}$ as that of the first example, except for $\tau = 1.0 \times 10^{-8}$ s. Both the innermost and outermost boundaries of the mesh are truncated by analytically known electric fields.




(a) (b)

Fig. 3.9. Simulation of a rectangular box with a hole discretized into tetrahedral elements: (a) Electric fields obtained from the proposed method and those from analytical results. (b) Entire solution error versus time for $\mathbf{E}$.

The proposed method takes 9.89 MB to store sparse matrices $\mathbf{S}_e$ and $\mathbf{S}_h$, and $2.7 \times 10^{-3}$ s to finish the simulation at one time step. We randomly select the 1-st and 8,612-th entries of vector $\{e\}$, and plot them in Fig. 3.9(a) in comparison with analytical solution. Excellent agreement can be observed. To assess the error of the entire $\{e\}$, we plot the entire solution error in Fig. 3.9(b), which again reveals good accuracy. In this example, we have also simulated to very late time to examine late-time stability. As can be seen from Fig. 3.10, the proposed method is stable.

### 3.3.4 Wave Propagation in a Tetrahedral Mesh of a Spherical Ring

This example is a spherical shells whose inner radius is 0.8 m, and outer radius is 1.2 m. It is discretized into tetrahedral elements in free space. The discretization

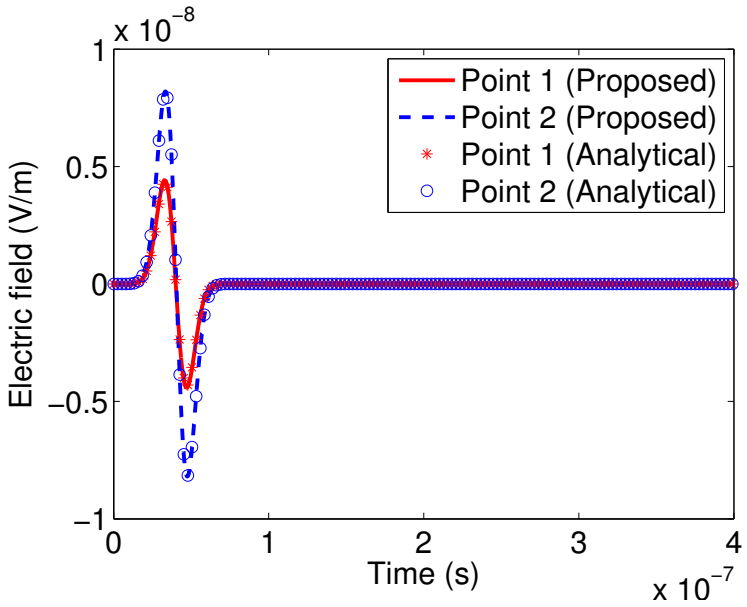


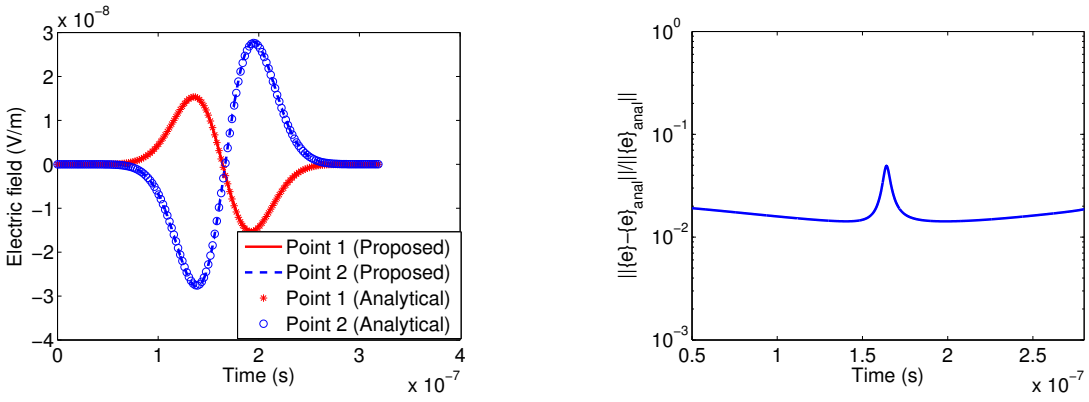

Fig. 3.10. Late-time simulation of a rectangular box with a hole.

results in 2,704 edges and 1,956 tetrahedrons. The aspect ratio of the tetrahedral mesh is 5.67. The incident $\mathbf{E}$ is the same as that of the first example, except for $\tau = 4.0 \times 10^{-8}$ s.

Analytically known electric fields are imposed to truncate the computational domain. The time step is chosen as $\Delta t = 2.0 \times 10^{-11}$ s. The proposed method takes 13.63 MB to store $\mathbf{S}_e$ and $\mathbf{S}_h$, and $3.6 \times 10^{-3}$ s to finish the simulation at one time step. In Fig. 3.11(a), we plot two electric field unknowns randomly selected from the entire $\{e\}$ vector, whose indices are 1 and 11,064. In Fig. 3.11(b), we plot the entire solution error shown in (3.41) with respect to time. Excellent agreement with analytical data can be observed from Fig. 3.11(a) and Fig. 3.11(b).

### 3.3.5 Lossy and Inhomogeneous Example Discretized into Triangular Prism Elements

Previous examples are all in free space. In this example, we simulate a structure with lossy conductors and inhomogeneous materials shown in Fig. 3.12. The structure




(a) (b)

Fig. 3.11. Simulation of a spherical ring discretized into tetrahedral elements: (a) Electric fields obtained from the proposed method as compared with analytical results. (b) Entire solution error versus time for **E**.

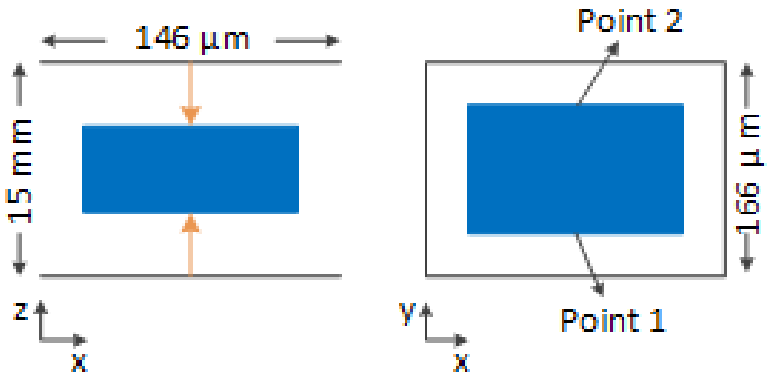
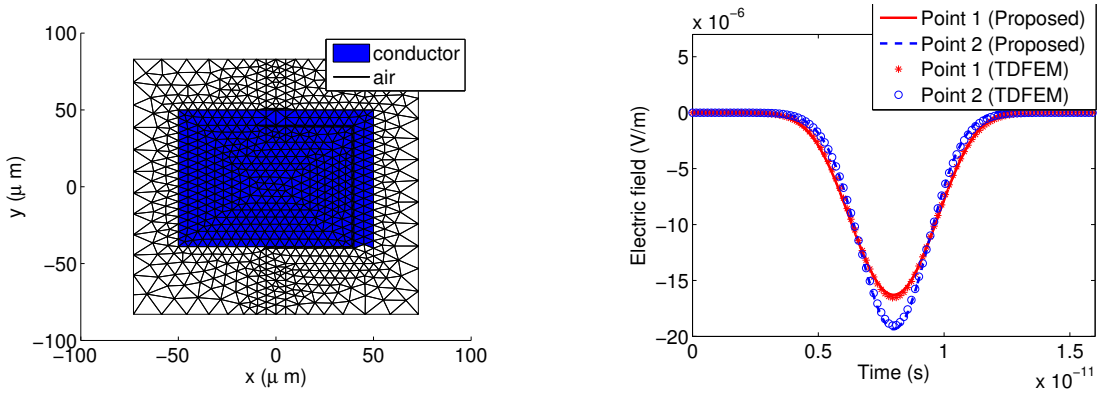



Fig. 3.12. Simulation of a lossy and inhomogeneous example discretized into triangular prism elements: Illustration of the structure.

is discretized into three layers of triangular prism elements. The thickness of each layer is 5 mm. The top view of the mesh is shown in Fig. 3.13(a). The discretization results in 12,574 triangular prism elements and 5,022 edges. A square conductor is located at the center of the second layer, which is shown in blue in Fig. 3.13(a). The metal conductivity is $5 \times 10^7$ S/m. The second layer is filled by a material of dielectric constant 4. The rest of the two layers have dielectric constant 1. The top and bottom



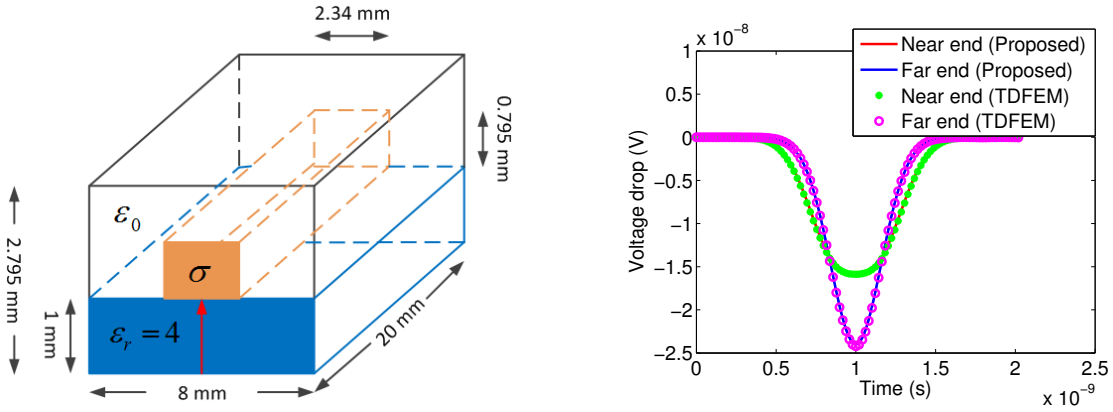

(a) (b)

Fig. 3.13. Simulation of a lossy and inhomogeneous example discretized into triangular prism elements: (a) Top view of the mesh. (b) Electric fields simulated from the proposed method as compared with the TDFEM results.

boundaries are truncated by perfect electric conducting (PEC) boundary condition, while perfect magnetic conductor (PMC) boundary condition is imposed on the other four sides. A current source with a Gaussian's derivative pulse is launched having $\tau = 2.0\times10^{-12}$ s. $\Delta t = 5.0\times10^{-16}$ s is chosen, since the smallest size has a micrometer dimension. The proposed method takes 0.12 GB to store sparse $\mathbf{S}_e$ and $\mathbf{S}_h$, and 0.10 s to finish the simulation at one time step. To examine the accuracy of the proposed method, we simulate the same example by using the TDFEM as the reference. Fig. 3.13(b) compares the simulated electric fields at two observation points located at the front and back end of the square conductor with those simulated by TDFEM. Excellent agreement is observed.

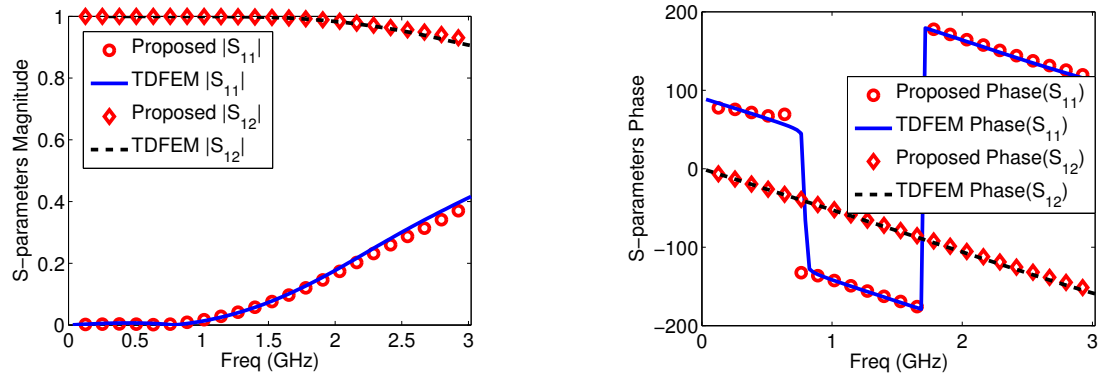
### 3.3.6 Lossy and Inhomogeneous Microstrip Line Discretized into Tetrahedral Elements

In this example, we simulate a 20-mm-long inhomogeneous and lossy microstrip line discretized into tetrahedral elements. The structure details can be found in Fig. 3.14(a). The aspect ratio of the tetrahedral mesh is 8.78. The substrate has a material




(a) (b)

Fig. 3.14. (a) Illustration of the microstrip line. (b) Voltages simulated from the proposed method in comparison with TDFEM results.




(a) (b)

Fig. 3.15. Simulation of a lossy and inhomogeneous microstrip line discretized into tetrahedral elements: (a) S-parameter Magnitude. (b) S-parameter Phase (Degrees).

of $\epsilon_r = 4$. The conductivity of the metal strip is $5.8 \times 10^7$ S/m. The discretization results in 35,283 edges and 28,365 tetrahedrons. A current source is imposed at the near end with $j = 2(t - t_0)\exp(-(\frac{t-t_0}{\tau})^2)$ and $\tau = 2.5 \times 10^{-10}$ s. The bottom plane is terminated with PEC, while PMC is applied to other boundaries. The time step used is $6.0 \times 10^{-14}$ s. The proposed method takes only 0.22 GB to store sparse $\mathbf{S}_e$

and $\mathbf{S}_h$, and 0.10 s to finish the simulation at one time step. The voltage between the microstrip and the ground plane at the near end ($z = 0$) and far end ($z = 20$ mm) is extracted, and compared with the reference TDFEM solution in Fig. 3.14(b). It is evident that the results obtained from the proposed method agree very well with the reference results. In Fig. 3.15, we plot the S-parameters extracted from the time-domain waveforms of the proposed method in comparison with those generated from TDFEM. Excellent agreement is observed in the entire frequency band simulated.

### 3.3.7 CPU Time and Memory Comparison

In this section, we simulate a large example to compare the performance of the proposed matrix-free method against the TDFEM which is equally capable of handling unstructured meshes, but not free of matrix solutions. This example is a circular cylinder of radius 1 m discretized into 25 layers of triangular prism elements. The incident field is a plane wave having a Gaussian's derivative pulse with $\tau = 10^{-8}$ s. An analytical absorbing boundary condition is imposed at the outermost boundary. The discretization results in 3,718,900 $\mathbf{E}$ unknowns using the zeroth-order TDFEM. A similar number of unknowns, 3,741,700 $\mathbf{E}$ unknowns, is generated in the proposed method for a fair comparison. Since TDFEM requires solving a mass matrix, we perform the LU factorization of the sparse mass matrix once before time marching, and use backward/forward substitution to obtain the solution at each time step. The TDFEM takes 2267.71 s and more than 72 GB memory to finish the factorization. This large memory cost is due to the fact that although the matrix being factorized is sparse, its $\mathbf{L}$ and $\mathbf{U}$ factors are generally dense. During time marching, the TDFEM costs 9.22 s at each time step. In contrast, since the proposed method is matrix-free, it does not need any memory as well as CPU time to factorize and solve the matrix. It takes only 5.2 GB memory to store the sparse $\mathbf{S}_e$ and $\mathbf{S}_h$, and 2.7 s for performing the time marching for one time step. Obviously, the proposed method significantly outperforms TDFEM in terms of computational efficiency. As for accuracy, the entire

solution error across the whole time window is $< 0.01\%$ for TDFEM and $0.05\%$ for the proposed method, as compared with the analytical result. Therefore, the proposed method can achieve a similar level of good accuracy as TDFEM. The difference in accuracy can be attributed to the difference in space as well as time discretizations of the two methods.

## 3.4 Conclusion

In this chapter, a new matrix-free time-domain method with a modified-basis formulation is developed for solving Maxwell's equations in general 3-D unstructured meshes. The method is naturally free of matrix solutions. No mass lumping is required, as the mass matrix is diagonal in nature by the proposed algorithm of discretizing Maxwell's equations. The method handles arbitrary unstructured meshes with the same ease as an FEM. It overcomes the absolute instability of an explicit method when an unsymmetrical operator having complex-valued and even negative eigenvalues is involved. Both stability and accuracy are theoretically guaranteed, and the tangential continuity of the fields is enforced at the material interfaces. It does not require dual mesh, projection, and interpolation. A set of modified vector basis functions are developed to directly connect the discretized Ampere's law with the discretized Faraday's law without any need for unknown transformation. Extensive numerical experiments on unstructured tetrahedral and triangular prism meshes, involving inhomogeneous, lossless, as well as lossy materials, have validated the accuracy, generality, and matrix-free property of the proposed method.

It is also worth mentioning that the proposed method can be flexibly extended to achieve any desired higher order accuracy by expanding one field unknown using arbitrary-order vector bases, and sampling the other field unknown in the loop orthogonal to the first field unknown in a higher order way. The modified higher order vector bases developed in this chapter can also be used in any other method that employs higher order bases. With these new bases, the relationship is explicitly known

between unknown fields and unknown coefficients of vector bases. The approach developed here and in [50] for stably simulating an unsymmetrical curl-curl operator can also be leveraged by the existing nonorthogonal FDTD methods for controlling stability.

# 4. MATRIX-FREE TIME-DOMAIN METHOD WITH TRADITIONAL VECTOR BASES IN UNSTRUCTURED MESHES

## 4.1 Introduction

In Chap. 2 and Chap. 3, 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. The traditional vector basis functions are modified appropriately to connect Faraday's law and Ampere's law together. In this chapter, we show that such a capability can be achieved with traditional 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.

## 4.2 Proposed Framework

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

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \tag{4.1}$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} + \sigma \mathbf{E} + \mathbf{J}, \tag{4.2}$$

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.

### 4.2.1 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 $\mathbf{N}_j$ $(j = 1, 2, ..., m)$ as the following

$$\mathbf{E} = \sum_{j=1}^{m} u_j \mathbf{N}_j, \tag{4.3}$$

where $u_j$ is the unknown coefficient of the $j$-th 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. This is different from many existing non-orthogonal FDTD methods, where the fields and fluxes are assigned only on the faces of the element.

Substituting the expansion of $\mathbf{E}$ into (4.1), computing $\mathbf{H}$ at $N_h$ points $\mathbf{r}_{hi}$ ($i = 1, 2, ..., N_h$), and then taking the dot product of the resultant with unit vector $\hat{h}_i$ at each point respectively, we obtain

$$\hat{h}_i \cdot \sum u_j \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}) = -\hat{h}_i \cdot \mu(\mathbf{r}_{hi}) \frac{\partial \mathbf{H}(\mathbf{r}_{hi})}{\partial t}, (i = 1, 2, ..., N_h) \tag{4.4}$$

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

$$\mathbf{S}_e\{u\} = -diag(\{\mu\}) \frac{\partial \{h\}}{\partial t}, \tag{4.5}$$

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

$$h_i = \mathbf{H}(\mathbf{r}_{hi}) \cdot \hat{h}_i, \tag{4.6}$$

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

$$\mathbf{S}_{e,ij} = \hat{h}_i \cdot \{\nabla \times \mathbf{N}_j\}(\mathbf{r}_{hi}), \tag{4.7}$$

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 $N_h \times N_e$. We loop over all elements to assemble $\mathbf{S}_e$. In each element, we build an elemental $\mathbf{S}_e$ matrix of size $n_h$ by $m$, where $n_h$ is the number of $\mathbf{H}$ points inside each element. The entries of elemental $\mathbf{S}_e$ are analytically known since bases $\mathbf{N}_j$ ($j = 1, 2, ..., m$) have analytical expressions. The elemental $\mathbf{S}_{e,ij}$ entries are then added upon the global $\mathbf{S}_e$ based on the global indexes of the local row index $i$, and column index $j$. Notice that 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. In addition, it is worth mentioning that different from the conventional assembling procedure of an FEM method where both rows and columns add, here the rows of $\mathbf{S}_e$ contributed by different elements do not add because each row corresponds to a different $\mathbf{H}$-unknown. However, the columns add, as the same tangential $\mathbf{E}$-unknown, i.e., an entry of $\{u\}$, can be shared by multiple elements. By using the same $\{u\}$ entry across elements, the tangential continuity of $\mathbf{E}$ is enforced.

### 4.2.2 Discretization of Ampere's Law

To discretize Ampere's law, we apply it at $\mathbf{r}_{ei}$ $(i = 1, 2, ..., N_e)$ 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 (4.3). We obtain

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = \epsilon(\mathbf{r}_{ei})\frac{\partial e_i}{\partial t} + \sigma(\mathbf{r}_{ei})e_i + \hat{e}_i \cdot \mathbf{J}(\mathbf{r}_{ei}), \tag{4.8}$$

where

$$e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i, \tag{4.9}$$

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 (4.8) is generated by using the $\mathbf{H}$ fields (obtained from (4.5)) 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

$$\mathbf{S}_h\{h\} = diag(\{\epsilon\})\frac{\partial\{e\}}{\partial t} + diag(\{\sigma\})\{e\} + \{j\}, \tag{4.10}$$

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

### 4.2.3 Connecting Ampere's Law to Faraday's Law

In order to connect (4.10) to (4.5), we need to find the relationship between $\{e\}$ and $\{u\}$. In Chap. 2 and Chap. 3, by making a minor modification of the traditional vector bases [48], 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 $1 \times 1$ or $2 \times 2$. The detailed formulation of $\mathbf{Q}$ will be given in next section.

In addition, when generating (4.5), 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 (4.5) to (4.10), we need to keep in mind that the $\mathbf{H}$-points and directions we choose should facilitate accurate generation of the $\{e\}$ desired in (4.5) so that we can march on in time step by step—from $\{e\}$ to $\{h\}$ via (4.5), and then from $\{h\}$ back to $\{e\}$ through (4.10).

### 4.2.4 Time Marching

A leap-frog-based time discretization of (4.5) and (4.10) clearly yields a time-marching scheme free of matrix solutions as follows:

$$\{h\}^{n+\frac{1}{2}} = \{h\}^{n-\frac{1}{2}} - diag(\{\frac{1}{\mu}\})\Delta t \mathbf{S}_e \mathbf{Q}\{e\}^n \tag{4.11}$$

$$\left(diag(\{\epsilon\}) + \frac{\Delta t}{2} diag(\{\sigma\})\right)\{e\}^{n+1} = \left(diag(\{\epsilon\}) - \frac{\Delta t}{2} diag(\{\sigma\})\right)\{e\}^n + \Delta t \mathbf{S}_h \{h\}^{n+\frac{1}{2}} - \Delta t \{j\}^{n+\frac{1}{2}}, \tag{4.12}$$

where $\Delta t$ is the time step, and the time instants for $\{e\}$ and $\{h\}$, denoted by superscripts, are staggered by half. Neither (4.11) nor (4.12) involves a matrix solution.

Equations (4.5) and (4.10) can also be solved in a second-order based way. Taking another time derivative of (4.10) and substituting (4.5), we obtain

$$\frac{\partial^2 \{e\}}{\partial t^2} + diag(\{\frac{\sigma}{\epsilon}\})\frac{\partial \{e\}}{\partial t} + \mathbf{S}\{e\} = -diag(\{\frac{1}{\epsilon}\})\frac{\partial \{j\}}{\partial t}, \tag{4.13}$$

where

$$\mathbf{S} = diag(\{\frac{1}{\epsilon}\})\mathbf{S}_h diag(\{\frac{1}{\mu}\})\mathbf{S}_e \mathbf{Q}. \tag{4.14}$$

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 (4.11) and (4.12) is the same as the central-difference-based explicit discretization of the second-order system (4.13). This can be readily seen by writing the counterpart of (4.12) for evaluating $\{e\}^n$, i.e., replacing $n$ by $n-1$ in (4.12), subtracting the resultant from (4.12), and then substituting (4.11) to replace the $\{h\}^{n+\frac{1}{2}} - \{h\}^{n-\frac{1}{2}}$ term. Since (4.11) and (4.12) are the same as the explicit discretization of (4.13), we can directly solve (4.13), which also has only half a number of unknowns. If $\{h\}$ unknowns are needed, they can readily be recovered from $\{e\}$ through (4.11).

### 4.2.5 Remark

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

## 4.3 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 are much simpler, 3-D formulations will be the focus of this section.

### 4.3.1 Accurate Construction of $\mathbf{S}_e$ and 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) [52]. These bases have constant tangential components along the edges where they are defined. The field represen-

tation 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 **E**, the resultant **H** is a constant in each element, and the **H** is only second-order accurate at the center point of each element. From such discrete **H**-fields, we cannot reversely obtain the **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

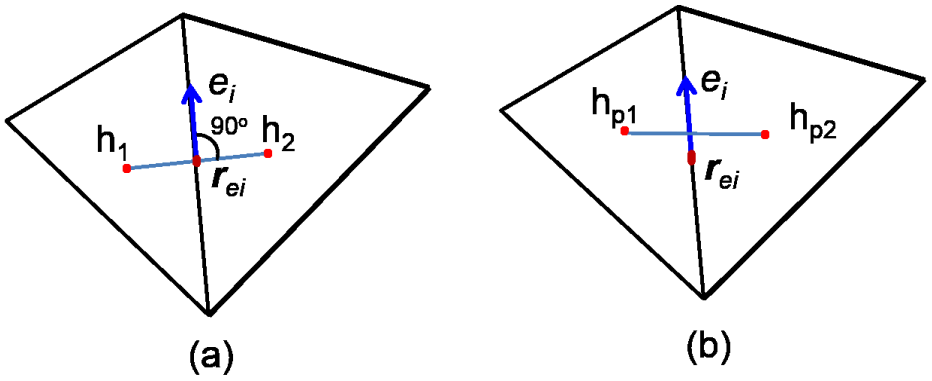



Fig. 4.1. (a) Locations of **H** points required for the accurate evaluation of $e$ at point $\mathbf{r}_e$. (b) Locations of **H** points with zeroth-order vector bases.

discretized into arbitrarily shaped triangular elements as an example. Consider an arbitrary $i$-th edge. With the zeroth-order vector bases to expand **E**, the $e_i$ shown in (4.9) has $\hat{e}_i$ the unit vector tangential to the $i$-th edge, and $\mathbf{r}_{ei}$ the center point of the $i$-th edge, as illustrated in Fig. 4.1. To obtain such an $e_i$ accurately from the discrete **H** (now $H_z$ only since the problem is 2-D), the two **H**-points should be located on the line that is perpendicular to the $i$-th edge and centered at the point $\mathbf{r}_{ei}$, as shown in Fig. 4.1(a). In this way, the edge is perpendicular to the **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 **H** unknowns. However, using the zeroth-order edge elements, the curl of **E** is constant in every element, thus we cannot generate **H** at the desired points accu-

rately. From another perspective, we can view the **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. 4.1(b).

To overcome the aforementioned problem, we propose to use higher-order curl-conforming vector bases to expand **E** in each element. With an order higher than zero, the curl of **E** and hence **H** is at least a linear function of $x$, $y$, and $z$ in each element. With this, the **H** can be obtained at an arbitrary point along an arbitrary direction accurately from (4.5). We hence can use this freedom to choose **H** points and directions in such a way that they can reversely generate **E** unknowns accurately from (4.10).

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. All the twenty first-order bases in a tetrahedral element together with their degrees of freedom defined in terms of locations $\mathbf{r}_{ei}$ and projection directions $\hat{e}_i$, $(i = 1, 2, ..., 20)$ are listed in Appendix A as well as Chap. 3 from Equ. (3.15) to Equ. (3.17). The vector bases for triangular prism element are listed in Appendix B. For other shaped elements, one can find the analytical expressions of higher-order vector bases from open literature.

### 4.3.2 Relationship between $\{u\}$ and $\{e\}$

The vector $\{u\}$ contains the unknown coefficients of vector basis functions as shown in (4.3), while vector $\{e\}$ contains the discrete electric fields at $\mathbf{r}_{ei}$ points along $\hat{e}_i$ directions as defined in (4.9). If $u_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i$, then $\{u\} = \{e\}$. Hence, (4.10) and (4.5) are directly connected to each other. Among higher-order vector basis

functions [43], 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 [48]. We can also keep the original higher-order vector bases as they are, but find the relationship between $\{u\}$ and $\{e\}$ as follows.

Substituting (4.3) into (4.9), we have

$$e_i = \mathbf{E}(\mathbf{r}_{ei}) \cdot \hat{e}_i = \sum u_j (\mathbf{N}_j(\mathbf{r}_{ei}) \cdot \hat{e}_i), \tag{4.15}$$

from which we obtain

$$\{e\} = \mathbf{P}\{u\}, \tag{4.16}$$

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

$$\mathbf{P}_{ij} = \mathbf{N}_j(\mathbf{r}_{ei}) \cdot \hat{e}_i. \tag{4.17}$$

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 $i$-th 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 $i$-th row of $\mathbf{P}$, and two nonzero elements on the $(i+1)$-th row of $\mathbf{P}$, forming a $2 \times 2$ diagonal block in $\mathbf{P}$ as the following

$$\mathbf{P}_i = \begin{bmatrix} 1 & \mathbf{N}_{i+1}(\mathbf{r}_{ei}) \cdot \hat{e}_i \\ \mathbf{N}_i(\mathbf{r}_{e,i+1}) \cdot \hat{e}_{i+1} & 1 \end{bmatrix}. \tag{4.18}$$

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

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & 0 & 0 & ... & 0 \\ 0 & \mathbf{P}_2 & 0 & ... & 0 \\ 0 & 0 & \mathbf{P}_3 & ... & 0 \\ 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & ... & \mathbf{P}_{...} \end{bmatrix}, \tag{4.19}$$

where each diagonal block $\mathbf{P}_i$ is equal to either 1 or a $2 \times 2$ matrix shown in (4.18), 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

$$\{u\} = \mathbf{Q}\{e\}. \tag{4.20}$$

The (4.5) hence can be rewritten as

$$\mathbf{S}_e\mathbf{Q}\{e\} = -diag(\{\mu\})\frac{\partial\{h\}}{\partial t}. \tag{4.21}$$

Thus, (4.10) and (4.5) are connected to each other.

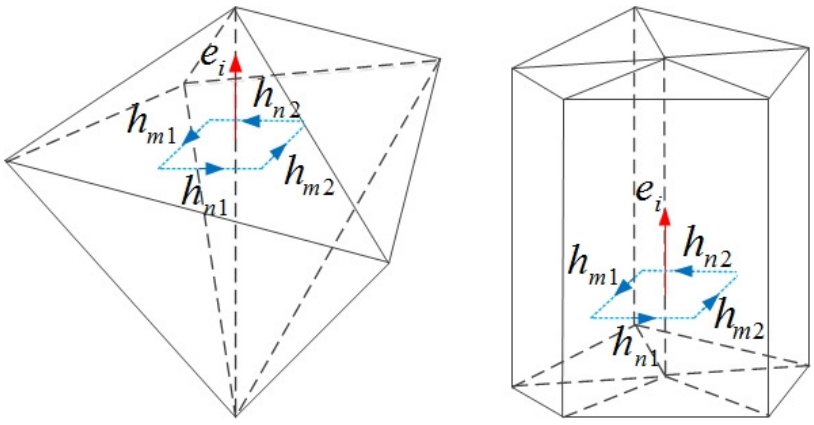



Fig. 4.2. $\mathbf{H}$ points and directions for generating $e_i$.

### 4.3.3 Accurate Construction of $\mathbf{S}_h$ and Choice of H's Points and Directions

To construct (4.10) accurately, intuitively, we can expand **H** by the same set of vector basis functions as that of **E** in each element. However, in this way, the degrees of freedom of **H** and those of **E** are located at the same points and along the same directions. From such a set of discrete **H**, it is not feasible to accurately obtain the desired **E** in (4.5). Our numerical experiments have also verified this fact. This is because the curl operation on **H** in each element will result in an **E** field whose space variation is one-order lower than **H**. Alternatively, we can test (4.2) by **E**'s vector bases and integrate over the computational domain. However, the resultant numerical system requires solving a matrix.

To construct a matrix-free solution and also with guaranteed accuracy, we propose to use an **H**-loop uniquely defined for each **E**'s degree of freedom to obtain the **E** desired in (4.5). 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. 4.2. 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 (4.5). 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 $\{\hat{h}_i\}$ $(i = 1, 2, ..., N_h)$.

With the aforementioned choice of **H**-points and directions, the $\hat{e}_i \cdot \nabla \times \mathbf{H}$ in (4.8) can be accurately discretized with second-order accuracy as the following

$$\hat{e}_i \cdot \{\nabla \times \mathbf{H}\}(\mathbf{r}_{ei}) = (h_{m1} + h_{m2})/l_{im} + (h_{n1} + h_{n2})/l_{in}, \tag{4.22}$$

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. 4.2. With (4.22), we obtain

$$\mathbf{S}_{h,ij} = 1/l_{ij}, \tag{4.23}$$

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 (4.5) accurately, due to the use of higher-order basis functions.

### 4.3.4 Imposing Boundary Conditions

The proposed method, in its first-order form (4.12) conforms to that of the FDTD numerical system; in its second-order form (4.13) 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 (4.5), 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 (4.10), since the $\{e\}$ entries at the $N_b$ points are known, the

updating of (4.10) 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 $\mathbf{E}$ 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 (4.13), 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 (4.13). To impose a PMC boundary condition, the total $\mathbf{E}$ unknown number is $N_e$ without any reduction. The (4.5) is formulated as it is since the $\mathbf{H}$-points having the PMC boundary condition can be placed outside the computational domain. As for (4.10), there is no need to make any change either since the tangential $\mathbf{H}$ 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 (4.5) and (4.10) 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.

In the framework and formulations described above, we expand the electric field into certain vector basis functions in each element, while sampling the magnetic field unknowns at discretized points along the loop individually defined for each $\mathbf{E}$'s degree of freedom. One can also switch the roles of the electric and magnetic fields: expand the magnetic field into vector basis functions in each element, while sampling the electric field unknowns along the loop defined for each magnetic field unknown. Which way to use depends on the convenience for solving a given problem.

## 4.4 Time Marching Free of Matrix-Solution with Guaranteed Stability

A leap-frog based time marching shown in (4.12) as well as a central-difference based time discretization of (4.13) 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 (4.12) and (4.13) [44]. The $z$-transform of the central-difference based time marching of (4.13), or (4.12) after eliminating $\{h\}$, results in the following equation:

$$(z-1)^2 + \Delta t^2 \lambda z = 0, \tag{4.24}$$

where $\lambda$ is the eigenvalue of $\mathbf{S}$. The two roots of (4.24) can be readily found as

$$z_{1,2} = \frac{2 - \Delta t^2 \lambda \pm \sqrt{\Delta t^2 \lambda (\Delta t^2 \lambda - 4)}}{2}. \tag{4.25}$$

If $\mathbf{S}$ is Hermitian positive semi-definite like that resulting from TDFEM or FDTD in an orthogonal grid, all its eigenvalues are non-negative real. Thus, we can always find a time step to make $z$ in (4.25) bounded by 1, and hence the explicit simulation of (4.13) as well as (4.12) 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 non-orthogonal FDTD methods, its eigenvalues either are real (can be negative) or come in complex-conjugate pairs. For complex-valued eigenvalues $\lambda$ as well as negative ones, the two roots $z_1$ and $z_2$ shown in (4.25) 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 (4.13) and (4.12) must be unstable. This fact was also made clear in [35]. For a general lossy problem, we can perform a similar stability analysis and find the same conclusion—if the $\mathbf{S}$ is not symmetric, a traditional explicit timed-domain simulation of (4.13) is absolutely 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 (4.13) [37]

$$\{e\}^{n+1} - 2\{e\}^n + \{e\}^{n-1} + \Delta t diag(\{\frac{\sigma}{\epsilon}\})(\{e\}^{n+1} - \{e\}^n) + \Delta t^2 \mathbf{S}\{e\}^{n+1} = -\Delta t^2 diag(\{\frac{1}{\epsilon}\}) \left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}, \tag{4.26}$$

where the $\{e\}$ associated with $\mathbf{S}$ is chosen at the $(n+1)$-th time step instead of the $n$-th step. Performing a stability analysis of (4.26) for lossless cases, we find the two roots of $z$ as

$$z_{1,2} = \frac{1}{1 \pm j\Delta t\sqrt{\lambda}}. \tag{4.27}$$

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 $\lambda$ can be complex-valued or even negative. To make the magnitude of (4.27) bounded by 1, we find that the time step needs to satisfy the following condition

$$\Delta t > 2\frac{|\mathrm{Im}(\sqrt{\lambda})|}{|\sqrt{\lambda}|^2}, \tag{4.28}$$

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 (4.26), we obtain

$$\tilde{\mathbf{D}}\{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} + \Delta t diag(\{\frac{\sigma}{\epsilon}\})\{e\}^n - \Delta t^2 diag(\{\frac{1}{\epsilon}\}) \left(\frac{\partial\{j\}}{\partial t}\right)^{n+1}, \tag{4.29}$$

where

$$\tilde{\mathbf{D}} = \mathbf{I} + \Delta t diag(\{\frac{\sigma}{\epsilon}\}) + \Delta t^2 \mathbf{S}. \tag{4.30}$$

Since $\tilde{\mathbf{D}}$ is not diagonal, (4.29) 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

$$\mathbf{D} = \mathbf{I} + \Delta t diag(\{\frac{\sigma}{\epsilon}\}). \tag{4.31}$$

Front multiplying both sides of (4.29) by $\mathbf{D}^{-1}$, we obtain

$$(\mathbf{I} + \tilde{\mathbf{M}})\{e\}^{n+1} = \mathbf{D}^{-1}\{f\}, \tag{4.32}$$

where $\{f\}$ is the right hand side of (4.29), and

$$\tilde{\mathbf{M}} = \Delta t^2 \mathbf{D}^{-1}\mathbf{S}. \tag{4.33}$$

Although (4.29) 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

$$\Delta t^2 < \frac{1}{\|\mathbf{S}\|}, \tag{4.34}$$

and hence

$$\Delta t^2 \|\mathbf{S}\| < 1. \tag{4.35}$$

This is because the time step for stability of a conventional central-difference based explicit simulation satisfies $\Delta t < 2/\sqrt{\rho(\mathbf{S})}$, where $\rho(\mathbf{S})$ is the spectral radius of $\mathbf{S}$. Although the $\mathbf{S}$ in the proposed method is different from that of the conventional TDFEM or FDTD, the matrix norm is similar since it represents the largest resonance frequency that can be numerically supported by a finite space discretization. This time step is also the time step required by accuracy when space step is determined by accuracy. Since $\mathbf{D}$ in (4.31) is diagonal, the norm of its inverse can be analytically evaluated as

$$\|\mathbf{D}^{-1}\| = 1/\min_{1 \le i \le N_e}(1 + \Delta t \sigma_i/\epsilon_i) = 1. \tag{4.36}$$

we hence obtain, from (4.35) and (4.36),

$$\|\tilde{\mathbf{M}}\| = \Delta t^2 \|\mathbf{D}^{-1}\mathbf{S}\| \le \Delta t^2 \|\mathbf{D}^{-1}\| \|\mathbf{S}\| < 1. \tag{4.37}$$

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

$$(\mathbf{I} + \tilde{\mathbf{M}})^{-1} = \mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \tilde{\mathbf{M}}^3 + \ldots, \tag{4.38}$$

which can be truncated after the first few terms without sacrificing accuracy due to (4.37). Thus, the system matrix has an explicit inverse, and hence no matrix solution is required in the proposed method. The final update equation becomes

$$\{e\}^{n+1} = (\mathbf{I} - \tilde{\mathbf{M}} + \tilde{\mathbf{M}}^2 - \cdots + (-\tilde{\mathbf{M}})^k)\mathbf{D}_i\{f\}, \tag{4.39}$$

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 (4.37) holds true, and the central-difference based time step (4.34) 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 (4.37), is essentially the square of the ratio of the actual time step used ($\Delta t$) 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 (4.38) is guaranteed and the convergence rate does not depend on the mesh quality. Notice that using (4.38) 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. It is also worth mentioning that the time step that violates (4.28) turns out to be small in the proposed method since the imaginary part of the complex eigenvalues is small as compared to the real part, owing to the accuracy of the proposed space discretization scheme.

The computational cost of (4.39) 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 (4.39) 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 (4.39) is the cost of multiplying $-\tilde{\mathbf{M}}$ by the vector obtained at the previous step, thus efficient.

## 4.5 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 **E**-**B** formulation like [53] 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 semi-definite. 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 $i$-th row of $\mathbf{S}_e$ by $\{e\}$ is nothing but

$$\frac{e_m - e_n}{W} - \frac{e_p - e_q}{L}, \tag{4.40}$$

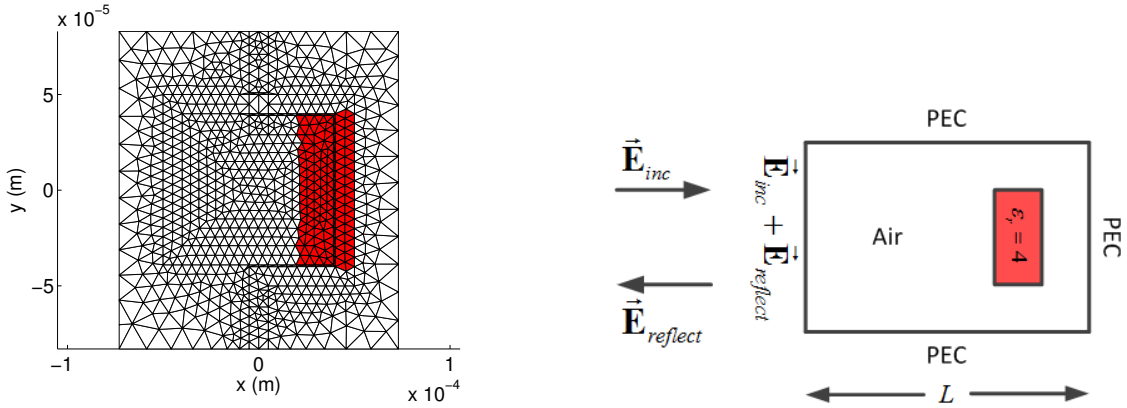
where $m$, $n$, $p$, $q$ are the global indexes of the four edge basis functions in the rectangular element where the **H** point is located, and $W$ and $L$ are the two side lengths of the rectangular element. It is evident that (4.40) is the same as that performed in the FDTD to produce the $H_z$ at the center of each **E**-loop. With $\mathbf{S}_h = \mathbf{S}_e^T$, the operation of $\mathbf{S}_h\{h\}$ is to do

$$\frac{1}{l_j}h_{p1} - \frac{1}{l_j}h_{p2} = \frac{h_{p1} - h_{p2}}{l_j}, \tag{4.41}$$

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 **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.

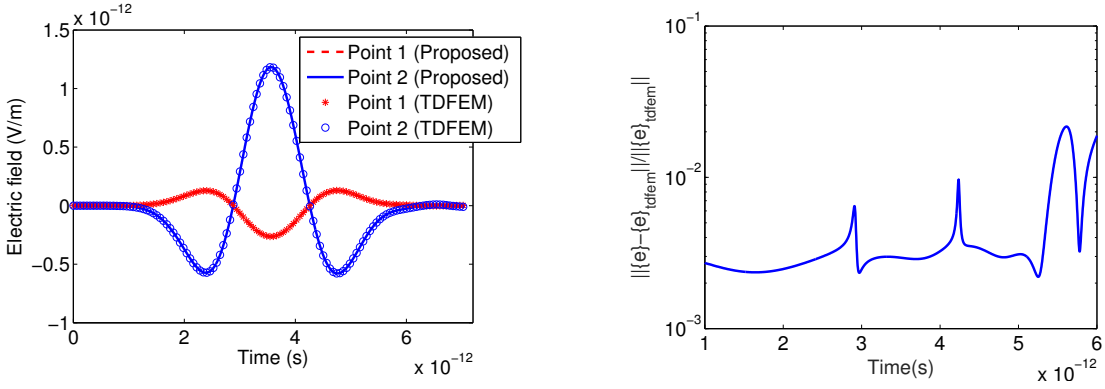



(a) (b)

Fig. 4.3. Simulation of wave propagation and reflection in a 2-D triangular mesh: (a) Mesh. (b) Illustration of incident wave and truncation boundary conditions.

## 4.6 Numerical Results

In this section, we simulate a variety of 2-D 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 comparison with both analytical solutions and the TDFEM method that is capable of handling unstructured meshes but not matrix-free.




(a) (b)

Fig. 4.4. Simulation of a 2-D triangular mesh: (a) Electric fields at two points. (b) *Entire* solution error v.s. time.

### 4.6.1 Wave Propagation and Reflection in a 2-D Triangular Mesh

The first example is a wave propagation and reflection problem in an 2-D triangular mesh shown in Fig. 4.3(a). Some mesh elements are very skewed due to fine features in a narrow gap whose size is less than a few $\mu$m. 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 $f(t) = 2(t - t_0)\exp(-(t - t_0)^2/\tau^2)$, $\tau = 8.0 \times 10^{-13}$ 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. 4.3(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 (4.38). 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 $5 \times 10^{-16}$ s. In Fig. 4.4(a), the electric fields at two points $\mathbf{r}_{p1} = (-5.912 \times 10^{-5}, -7.131 \times 10^{-5}, 0)$ m and $\mathbf{r}_{p2} = (-6.325 \times 10^{-5}, -6.434 \times 10^{-5}, 0)$ m 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. 4.4(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

$$\text{Error}_{entire}(t) = \frac{||\{e\}_{this}(t) - \{e\}_{ref}(t)||}{||\{e\}_{ref}(t)||}, \tag{4.42}$$

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

### 4.6.2 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. 4.5. 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 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 (4.38). In Fig. 4.6(a), we plot the first and 1,832-th entry randomly selected from the unknown $\{e\}$ vector, which represent $\mathbf{E}(\mathbf{r}_{ei})\cdot\hat{e}_i$, with $i = 1$, and 1,832

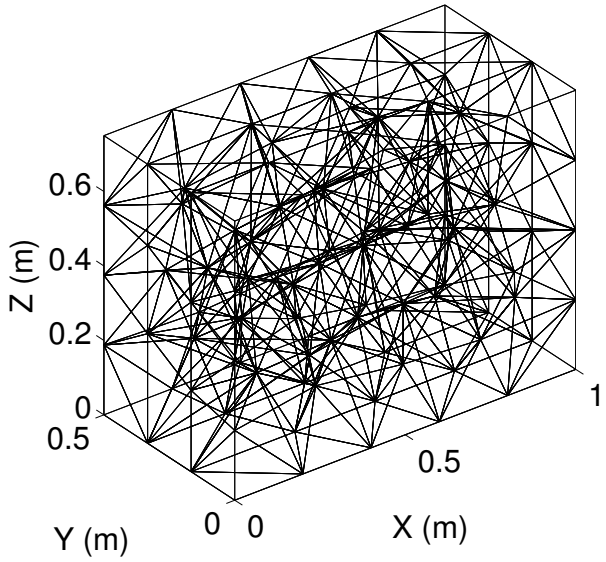
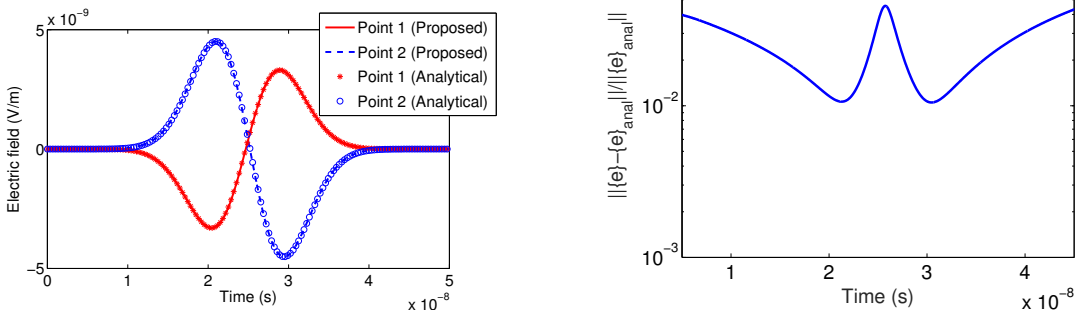



Fig. 4.5. Illustration of the tetrahedron mesh of a 3-D structure.

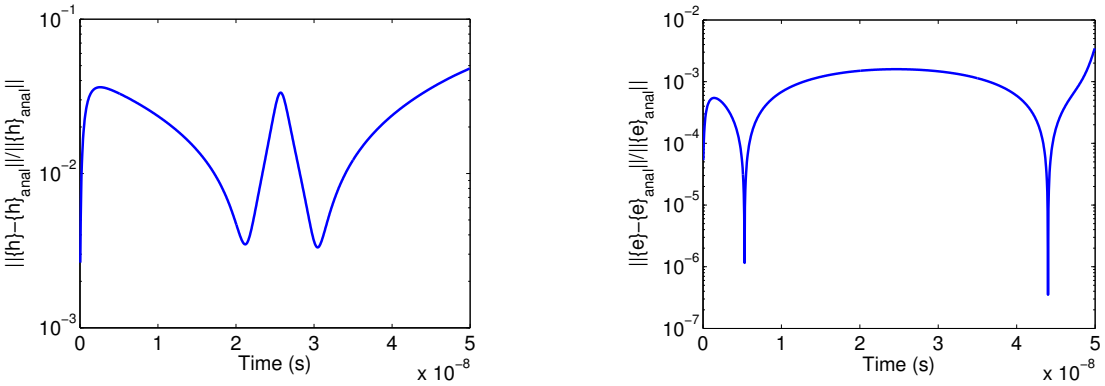
respectively. From Fig. 4.6(a), it can be seen clearly that the electric fields solved from the proposed method have an excellent agreement with analytical results.




(a) (b)

Fig. 4.6. 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 **E** unknowns v.s. time.

To further verify the accuracy of the proposed method in the entire computational domain, we assess the entire solution error (4.42) as a function of time, where the reference solution is analytical result $\{e\}_{anal}(t)$. In Fig. 4.6(b), we plot $\text{Error}_{entire}(t)$ across the 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. 4.6(b) is due to a comparison with close to zero fields.




(a) (b)

Fig. 4.7. (a) Entire solution error v.s. time of all **H** unknowns obtained from $\mathbf{S}_e$-rows of equations. (b) Entire solution error v.s. time of all **E** obtained from $\mathbf{S}_h$-rows of equations.

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 (4.5) only without (4.10) by providing an analytical $\{e\}$ to (4.5) at each time step. The resultant $\{h\}$ is then compared to analytical $\{h\}_{anal}$ at each time step. As can be seen from Fig. 4.7(a) where the following **H**-error

$$\frac{||h(t) - h_{anal}(t)||}{||h_{anal}(t)||} \tag{4.43}$$

is plotted with respect to time, the error of all **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 (4.10) only without (4.5) by providing an analytical $\{h\}$ to (4.10) at each time step. The relative error of all $\mathbf{E}$ unknowns shown in (4.42) as compared to analytical solutions is plotted with time in Fig. 4.7(b). Again, very good accuracy is observed across the whole time window, verifying the accuracy of $\mathbf{S}_h$.

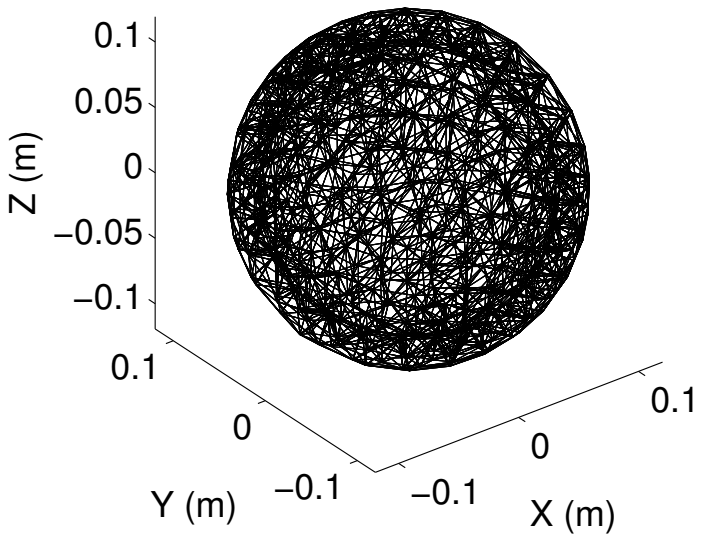
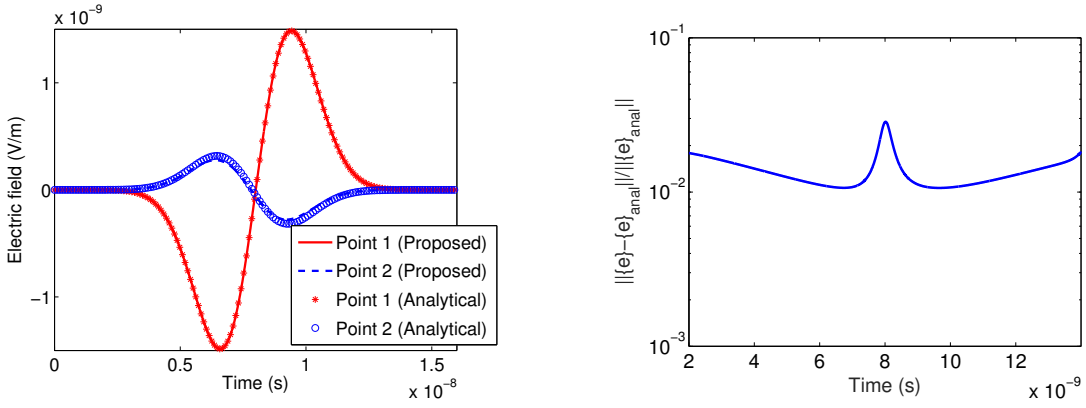



Fig. 4.8. Illustration of the tetrahedron mesh of a sphere structure.

### 4.6.3 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. 4.8. The discretization results in 3,183 edges and 1,987 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




(a) (b)

Fig. 4.9. 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 **E** unknowns v.s. time.

mesh is truncated by analytical **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 (4.38). The two degrees of freedom of the electric field, whose indices in vector $\{e\}$ are 1 and 9,762 respectively, are plotted in Fig. 4.9(a) in comparison with analytical data. Excellent agreement can be observed. In Fig. 4.9(b), we plot the entire solution error shown in (4.42) 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 accurate at all points in the computational domain for all time instants simulated.

### 4.6.4 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. 4.10. The structure has two layers of triangular prism elements (into the paper) with each layer being 0.05 m thick. The discretization results in 3,092 edges and 1,038 triangular prisms. Both the innermost and outermost boundaries are terminated by exact absorbing boundary condition, which is the analytical tangential

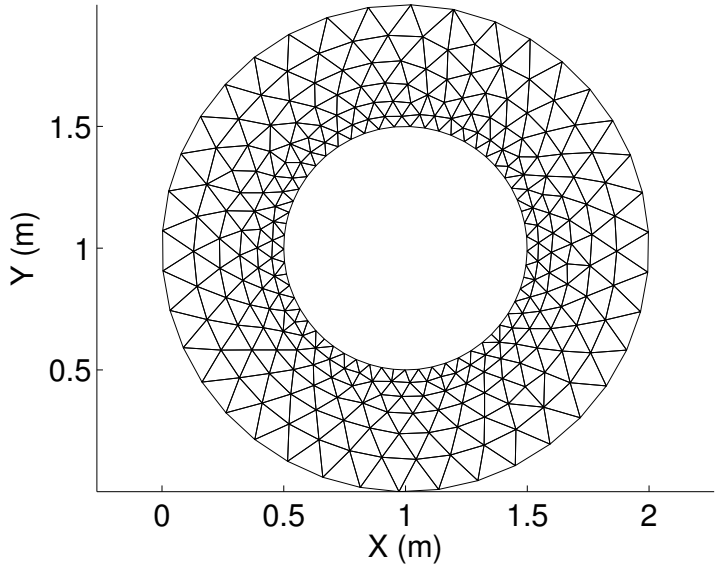
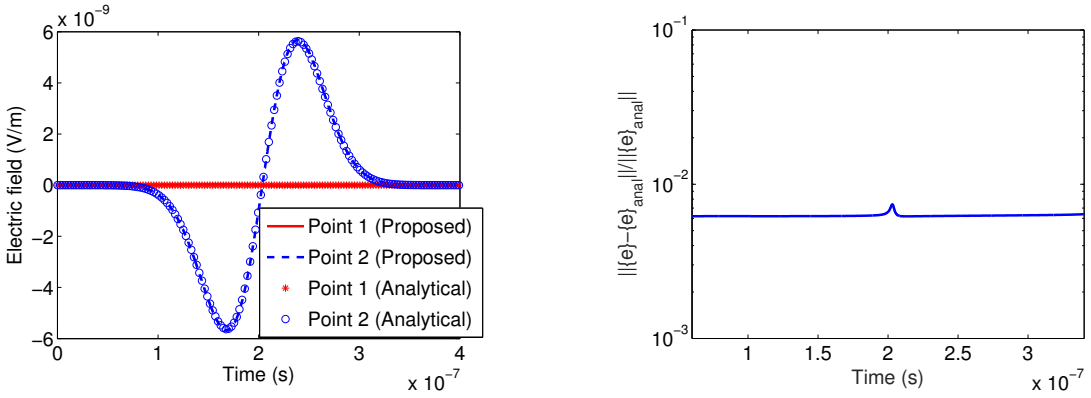



Fig. 4.10. Top view of the triangular prism mesh of an coaxial cylinder structure.




(a) (b)

Fig. 4.11. 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 **E** unknowns v.s. time.

**E** on the boundary. The incident **E** has the same form as that in the first example, but with $\tau = 5.0 \times 10^{-8}$ s. The $\Delta t$ used is $2.0 \times 10^{-11}$ 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. 4.11(a). Excellent agreement with analytical solutions can be observed. In Fig. 4.11(b), we plot the entire solution error shown in (4.42) 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.

### 4.6.5 Mesh with Mixed Elements

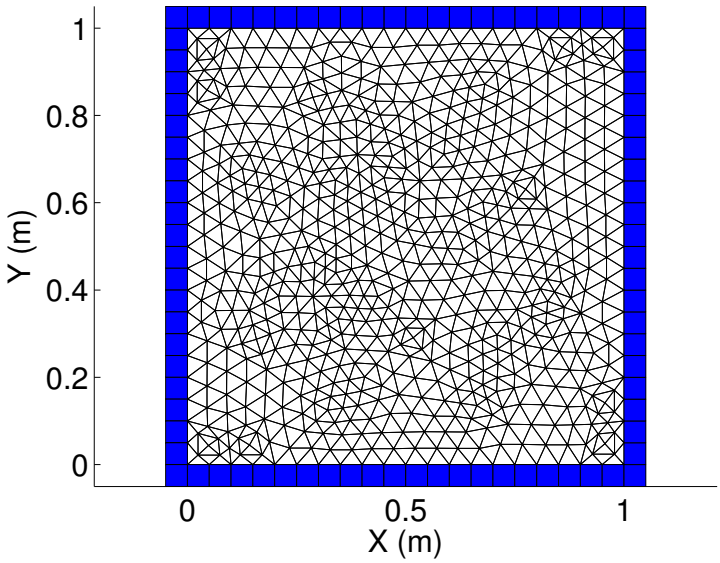
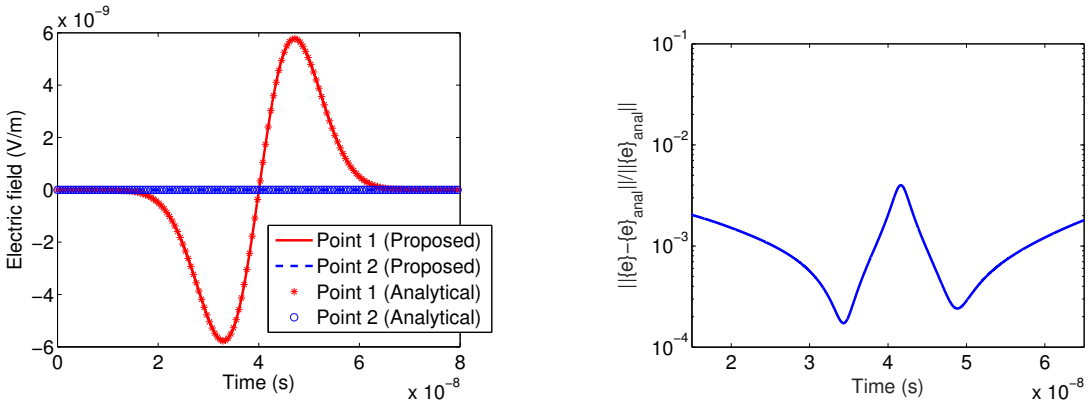



Fig. 4.12. Illustration of a mesh having different types of elements.

We have examined the capability of the proposed method in handling meshes made of different types of elements. This mesh is illustrated in Fig. 4.12, which consists of 1,312 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 twelve 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 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. 4.13(a), the electric fields at two randomly selected points are plotted in comparison with analytical data. Excellent agreement can be observed. In Fig. 4.13(b), 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.




(a) (b)

Fig. 4.13. Simulation of a mesh having different types of elements: (a) Two electric fields in comparison with analytical results. (b) Entire solution error for all **E** unknowns v.s. time.

### 4.6.6 S-parameter Extraction of a Lossy Package Inductor

In this example, we simulate a package inductor made of lossy conductors of conductivity $5.8 \times 10^7$ S/m, and embedded in a dielectric material of relative permittivity 3.4. Its geometry and material parameters are illustrated in Fig. 4.14. 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 $\mu$m from bottom to top, respectively. The top view of the mesh is shown in Fig. 4.15(a). The boundary conditions are PEC on the top

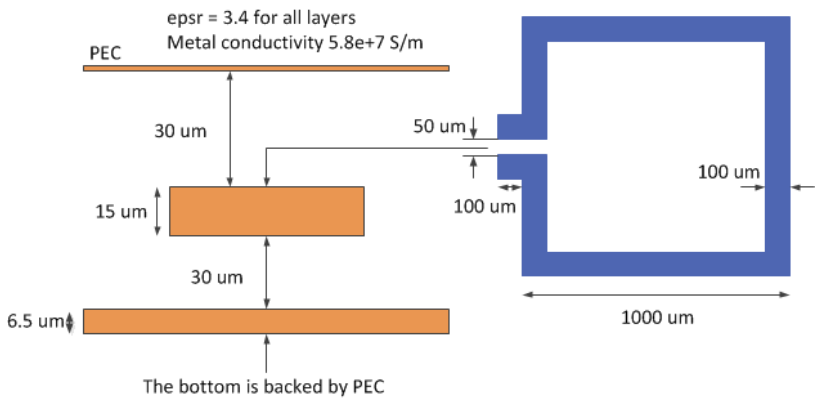


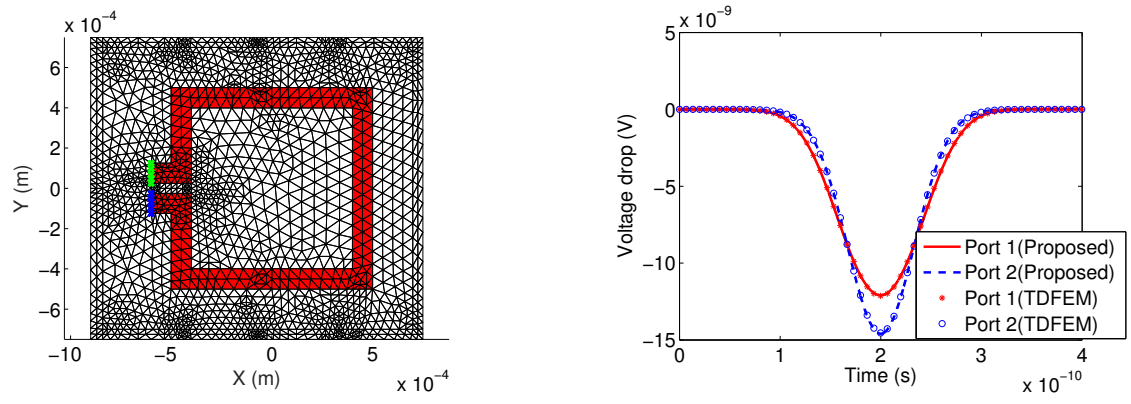

Fig. 4.14. Illustration of materials and geometry of a package inductor.

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. 4.15(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. 4.16 across the entire frequency band.

### 4.6.7 CPU Time and Memory Comparison

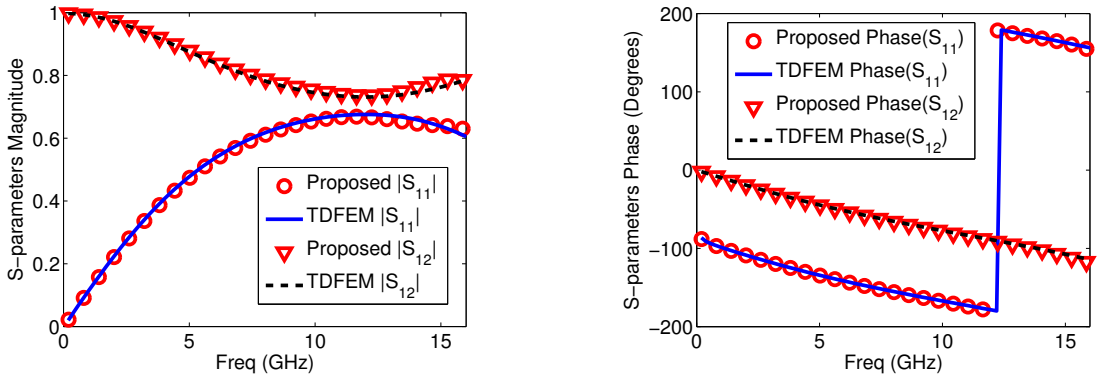
Among existing time-domain methods for handling unstructured meshes, the TDFEM only requires a single mesh like the proposed method. The TDFEM also has guaranteed stability and accuracy, and it ensures the tangential continuity of the fields across material interfaces. We hence choose the TDFEM to benchmark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




(a) (b)

Fig. 4.15. 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.




(a) (b)

Fig. 4.16. Simulation of a 3-D package inductor with dielectrics and lossy conductors: (a) Magnitude of S-parameters. (b) Phase of S-parameters.

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 9 in (4.38).

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 **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 central-difference-based TDFEM only requires solving a mass matrix, which is sparse and simple, its **L** and **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 **L** and **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 **L** and **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 [54]. 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 (4.42) is shown to be less than $4 \times 10^{-4}$ across the entire time window. The entire 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 **E** and **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.

## 4.7 Conclusion

In this chapter, 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. Different from the method developed in Chap. 2 and Chap. 3, the formulations presented in this chapter 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.

# 5. MATRIX-FREE TIME-DOMAIN METHOD WITH UNCONDITIONAL STABILITY IN UNSTRUCTURED MESHES

## 5.1 Introduction

A matrix-free method does not require the solution of a system matrix. Hence, it has a great potential of solving large-scale problems. An explicit FDTD method is free of matrix solutions [2]. Its stability limit has also been overcome by advanced research. However, the method is only applicable to an orthogonal grid. Various work has been done to extend the FDTD to unstructured meshes. In Chap. 2, 3 and 4, a matrix-free time-domain method is developed for both 2-D and 3-D scenarios. This method is independent of the element shape used for discretization [50]. Its accuracy and stability are shown to be satisfactory. Nevertheless, the method's time step is still restricted by the smallest space step.

Unlike the curl-curl operator of an FDTD method, which is symmetric and positive semi-definite, the curl-curl operator resulting from a matrix-free method is, in general, unsymmetrical in an unstructured mesh. Such an operator can support complex-valued and negative eigenvalues. They would even make a traditional explicit time marching absolutely unstable. Hence, it is challenging to further enlarge the time step of a matrix-free method in an unstructured mesh to any desired value. In this chapter, we overcome this challenge and successfully develop an unconditionally stable matrix-free method applicable to arbitrarily shaped unstructured meshes. As a result, the advantages of a matrix-free method in time domain are accentuated, while its shortcoming in time step is remedied, permitting an efficient analysis of large-scale and multi-scale problems. Numerical experiments have demonstrated the accuracy and efficiency of the proposed method.

## 5.2 Proposed Method

Consider a general electromagnetic problem discretized into arbitrarily shaped elements. Based on [50], the Faraday's law and Ampere's law can be discretized into the following forms:

$$\mathbf{S}_e\{e\} = -diag(\{\mu\})\frac{\partial\{h\}}{\partial t}, \tag{5.1}$$

$$\mathbf{S}_h\{h\} = diag(\{\epsilon\})\frac{\partial\{e\}}{\partial t} + \{j\}, \tag{5.2}$$

where $\{e\}$ is a global vector containing $N_e$ electric field unknowns, and $\{h\}$ is a global vector containing $N_h$ magnetic field unknowns. The $\mathbf{S}_e\{e\}$ represents the discretized $\nabla \times \mathbf{E}$, while $\mathbf{S}_h\{h\}$ describes the discretized $\nabla \times \mathbf{H}$. Both $\mathbf{S}_h$ and $\mathbf{S}_e^T$ are sparse matrices of $N_e \times N_h$ size. The $diag(\{\mu\})$ and $diag(\{\epsilon\})$ are diagonal matrices containing the permittivity and conductivity, and $\{j\}$ denotes a current source vector. In each element, $\mathbf{E}$ is expanded into higher-order bases, and hence the $\{h\}$ obtained from (5.1) is accurate at any point along any direction. The $\{h\}$ is then chosen along the orthogonal loops defined for each $\mathbf{E}$ unknown. The accuracy of (5.2) is thus guaranteed as well.

If we eliminate $\{h\}$ from (5.1) and (5.2), we can obtain the following second-order equation for $\{e\}$

$$\frac{\partial^2\{e\}}{\partial t^2} + \mathbf{S}\{e\} = -diag\left(\left\{\frac{1}{\epsilon}\right\}\right)\frac{\partial\{j\}}{\partial t}, \tag{5.3}$$

where $\mathbf{S} = diag(\{\frac{1}{\epsilon}\})\mathbf{S}_h diag(\{\frac{1}{\mu}\})\mathbf{S}_e$. Since $\mathbf{S}$ is unsymmetrical supporting complex-valued eigenvalues, a brute-force central-difference based time marching of (5.3), though free of matrix solutions, is absolutely unstable. This problem was circumvented by resorting to a backward-difference discretization but using a central-difference-based time step. Since this time step satisfies $\Delta t < 1/\sqrt{\rho(\mathbf{S})}$, where $\rho(\mathbf{S})$ denotes the spectral radius of $\mathbf{S}$, the matrix resulting from the backward difference has an explicit inverse. Thus, no matrix solution is needed. However, this also makes the time step depend on space step. Next, we first present the proposed method for solving this problem, and then explain how it works.

*Method:* Let the eigenvalues of $\mathbf{S}$ be $\xi_i$ $(i = 1, 2, \ldots, N_e)$. The theoretical value of the smallest one is zero since $\mathbf{S}$ has a nullspace. Given any time step $\Delta t$ no matter how large it is, the $\xi_i$ can be partitioned into two groups. One satisfies $\Delta t < 1/\sqrt{|\xi_i|}$, while the other does not. It is the latter that prevents a matrix-free time marching of (5.3). Let their corresponding eigenmodes be $\mathbf{U}_h$. These modes clearly have the largest eigenvalues of $\mathbf{S}$. Unlike those in FDTD, the eigenvectors of $\mathbf{S}$ are not orthogonal since $\mathbf{S}$ is not symmetric. We hence orthogonalize $\mathbf{U}_h$ to obtain $\mathbf{V}_h$. We then upfront change the system matrix $\mathbf{S}$ to $\mathbf{S}_l$ as follows

$$\mathbf{S}_l = \mathbf{S} - \mathbf{V}_h \mathbf{V}_h^H \mathbf{S}, \tag{5.4}$$

and perform a time marching on the updated new system $\mathbf{S}_l$

$$\frac{\partial^2 \{e\}}{\partial t^2} + \mathbf{S}_l \{e\} = -diag\left(\left\{\frac{1}{\epsilon}\right\}\right) \frac{\partial \{j\}}{\partial t}. \tag{5.5}$$

The above can be proved to have all eigenvalues satisfying $\sqrt{|\xi_i|} < 1/\Delta t$ (to be given in next subsection), and hence its time marching is free of matrix solutions for the given time step no matter how large it is. After obtaining $\{e\}^{n+1}$ from (5.5) at every step, we add the following treatment

$$\{e\}^{n+1} = \{e\}^{n+1} - \mathbf{V}_h \mathbf{V}_h^H \{e\}^{n+1} \tag{5.6}$$

to ensure the solution is free of $\mathbf{V}_h$-modes.

The complete solution $\{e\}$ can be expanded as $\{e\} = \{e_h\} + \{e_l\} = \mathbf{V}_h\{y_h\} + \mathbf{V}_l\{y_l\}$, where $\mathbf{V}_l$ is orthogonal to $\mathbf{V}_h$. Using the aforementioned procedure, we find $\{e_l\}$. To find $\{e_h\}$, we front multiply (5.3) by $\mathbf{V}_h^H$, obtaining

$$\frac{\partial^2 \{y_h\}}{\partial t^2} + \mathbf{Q}\{y_h\} = \{b\}, \tag{5.7}$$

where $\{b\} = \mathbf{V}_h^H \left(-diag\left(\left\{\frac{1}{\epsilon}\right\}\right) \frac{\partial \{j\}}{\partial t} - \mathbf{S}\{e_l\}\right)$, and $\mathbf{Q} = \mathbf{V}_h^H \mathbf{S} \mathbf{V}_h$. This is a small system of equations, whose size is $k$ (the number of $\mathbf{V}_h$ modes). It can further be transformed to a diagonal system of

$$\frac{\partial^2 \{w\}}{\partial t^2} + \mathbf{\Lambda}_q \{w\} = \{f\}, \tag{5.8}$$

where $\mathbf{\Lambda}_q$ is diagonal containing eigenvalues of $\mathbf{Q}$. After solving (5.8), we can obtain $\{e_h\} = \mathbf{V}_h\mathbf{V}_r\{w\}$, where $\mathbf{V}_r$ is the eigenvector matrix of small $\mathbf{Q}$ matrix. The total solution at each time step can then be obtained as $\{e\} = \{e_l\} + \{e_h\}$. Since $k$ is much smaller than $N_e$, the time cost of this step is trivial. In addition, the time step used for simulating the diagonal system (5.10) can be arbitrarily large with a backward difference.

*How It Works:* The field solution obtained from the proposed method is the same as that of (5.3). To prove, we can substitute $\{e\} = \mathbf{V}_h\{y_h\} + \mathbf{V}_l\{y_l\}$ into (5.3) , and multiply the resultant by $\mathbf{V}_l^H$. Since $\mathbf{V}_h$ is orthogonalized from eigenvector matrix $\mathbf{U}_h$, $\mathbf{V}_l^H\mathbf{S}\mathbf{V}_h = 0$ holds true. We hence obtain

$$\frac{\partial^2\{y_l\}}{\partial t^2} + \mathbf{V}_l^H\mathbf{S}\{u_l\} = \mathbf{V}_l^H\{b\}. \tag{5.9}$$

Multiplying both sides by $\mathbf{V}_l$, and recognizing $\mathbf{V}_l\mathbf{V}_l^H = \mathbf{I} - \mathbf{V}_h\mathbf{V}_h^H$, we obtain

$$\frac{\partial^2\{u_l\}}{\partial t^2} + (\mathbf{I} - \mathbf{V}_h\mathbf{V}_h^H)\mathbf{S}\{u_l\} = (\mathbf{I} - \mathbf{V}_h\mathbf{V}_h^H)\{b\}, \tag{5.10}$$

the solution of which is the same as those obtained from (5.5) and (5.6). The second step of the proposed method is to find $\{y_h\}$, thereby $\{e_h\}$. Hence, it is evident that the proposed method solves (5.3) without any approximation. It is worth mentioning that to make an FDTD stable, the second step can be saved since the eigenvectors are orthogonal, and only $\mathbf{U}_l$ is required for accuracy. Here, $\mathbf{U}_l$ can have a small projection onto $\mathbf{U}_h$. Therefore, some of the eigenmodes of (5.8) may not be ignored.

Now, we shall prove why $\mathbf{S}_l$ permits the use of any desired time step. Let the eigenvectors of $\mathbf{S}$ be $\mathbf{U} = [\mathbf{U}_h, \mathbf{U}_l]$. Since $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$, and $\mathbf{V}_l^H\mathbf{U}_h = 0$, $\mathbf{S}_l$ can be written as

$$\begin{aligned}\mathbf{S}_l &= \mathbf{V}_l\mathbf{V}_l^H\mathbf{S} = \mathbf{V}_l\mathbf{V}_l^H\left[\mathbf{U}_h\Lambda_h(\mathbf{U}^{-1})_h + \mathbf{U}_l\Lambda_l(\mathbf{U}^{-1})_l\right] \\ &= \mathbf{V}_l\mathbf{V}_l^H\left[\mathbf{U}_l\Lambda_l(\mathbf{U}^{-1})_l\right] = \mathbf{V}_l\mathbf{V}_l^H\left[\mathbf{U}diag\{0, \Lambda_l\}(\mathbf{U}^{-1})\right],\end{aligned}$$

where $(\mathbf{U}^{-1})_{h/l}$ denotes the rows of $\mathbf{U}^{-1}$ corresponding to the $\Lambda_{h/l}$ part. The spectral radius of $\mathbf{S}_l$ is hence bounded by that of $\Lambda_l$, which satisfies $\sqrt{|\xi_i|} < 1/\Delta t$.

*Computational Efficiency:* The number of $\mathbf{V}_h$, $k$, is in general not large, as it is proportional to the number of fine elements. In addition, since $\mathbf{V}_h$'s eigenvalues are the largest of $\mathbf{S}$, they can be efficiently found in $O(k^2 N_e)$ operations. Moreover, $\mathbf{V}_h$ is time independent. Once found, it can be reused for different simulations.

## 5.3 Numerical Results

### 5.3.1 Wave Propagation in 2-D Triangular Mesh

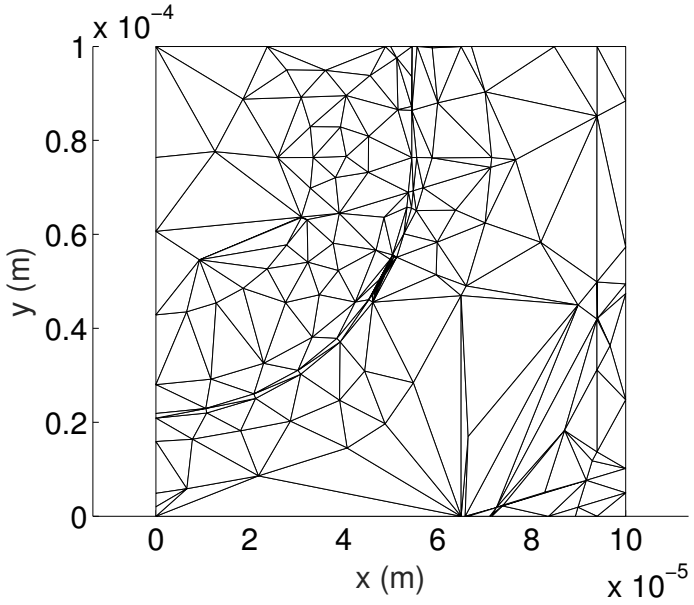



Fig. 5.1. Illustration of a 2D domain with a triangular mesh.

We first simulate a free-space wave propagation problem in a 2-D triangular mesh. This mesh is highly irregular as illustrated in Fig. 5.1. The incident electric field $\mathbf{E} = \hat{y} f(t - x/c)$ where $f(t) = 2(t - t_0)e^{-(t-t_0)^2/\tau^2}$ with $t_0 = 4\tau$ and $\tau = 8 \times 10^{-13}$ s. An analytical absorbing boundary condition is applied on the outermost boundary. The proposed method is able to use a time step of $2.0 \times 10^{-14}$ s, which is solely determined by accuracy. In contrast, the reference method [50] has to use a time step of $1.17 \times 10^{-17}$ s. In Fig. 5.2, we plot the entire solution error measured by

$||e - e_{ref}||/||e_{ref}||$ as a function of time, where $e_{ref}$ is the solution obtained from the reference method, while $e$ is the solution of the proposed method. It is evident that the proposed method is accurate at all points and across the entire time window simulated. In Fig. 5.3, we plot the field waveforms randomly selected at two points. They show excellent agreement with the reference results. The proposed method takes only 12.745 s to finish the entire simulation from finding $\mathbf{V}_h$ to the matrix-free time marching, whereas the reference method takes 260.174 s.

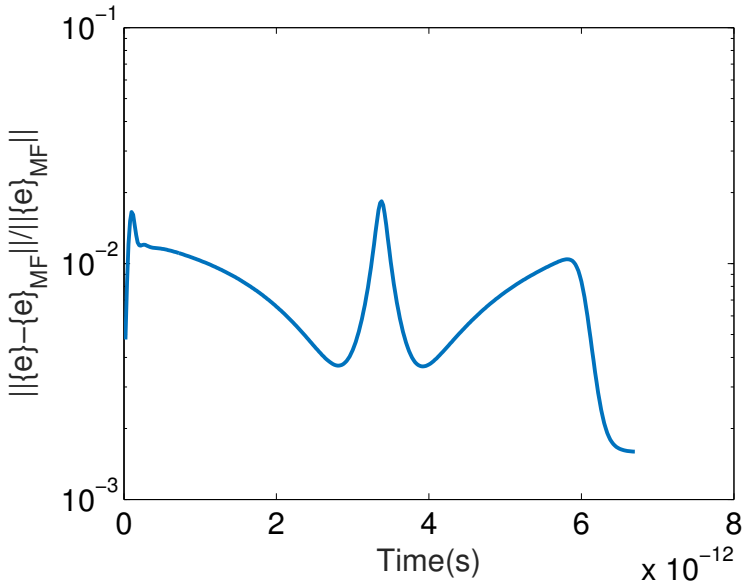



Fig. 5.2. Simulation of a 2D domain with a triangular mesh: *Entire* solution error v.s. time.

### 5.3.2 Wave Propagation in 3-D Tetrahedral Mesh

The second example is a 3-D cube of $0.5 \times 0.5 \times 0.405$ m$^3$ discretized into tetrahedron elements. The smallest space step is 0.005 m while the largest one is 0.1 m. The incident wave is the same as that in the first example but with $\tau = 2 \times 10^{-9}$ s. With 690 $\mathbf{V}_h$-modes removed, the time step is increased from $2.9 \times 10^{-13}$ s to the one required by accuracy, which is $3.0 \times 10^{-11}$ s. As seen from Fig. 5.4 the simulated

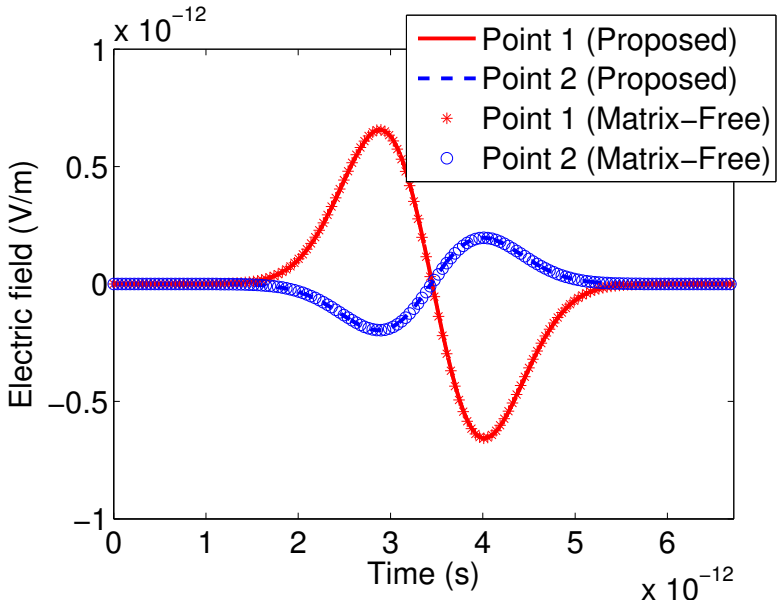



Fig. 5.3. Simulation of a 2D domain with a triangular mesh: electric field at observation points.

fields agree very well with the reference results. The total simulation time of the proposed method is 38.844 s including every step, in contrast to the 153.514 s cost by the reference method.

### 5.3.3 Simulation of a Parallel Plate

Finally, we simulate a 3-D parallel plate excited by a current source. The mesh details are shown in Fig. 5.5, and it involves 350 tetrahedral elements and 544 edges. The current source is launched along the green line shown in Fig. 5.5. Its expression is $\mathbf{J} = \hat{z}2(t - t_0)\exp -(t - t_0)^2/\tau^2$ with $\tau = 1$ s and $t_0 = 4\tau$. The matrix-free time-domain method requires the time step to be less than $2.4 \times 10^{-11}$ s to guarantee stability. This renders an estimated total CPU time $2.0104 \times 10^8$ s to finish the simulation. It's impossible to run such a long time to obtain the solution. For convenience, we can find out the voltage drop between the two PEC plates analytically since the input frequency is very low. In that case, the parallel plate can be viewed

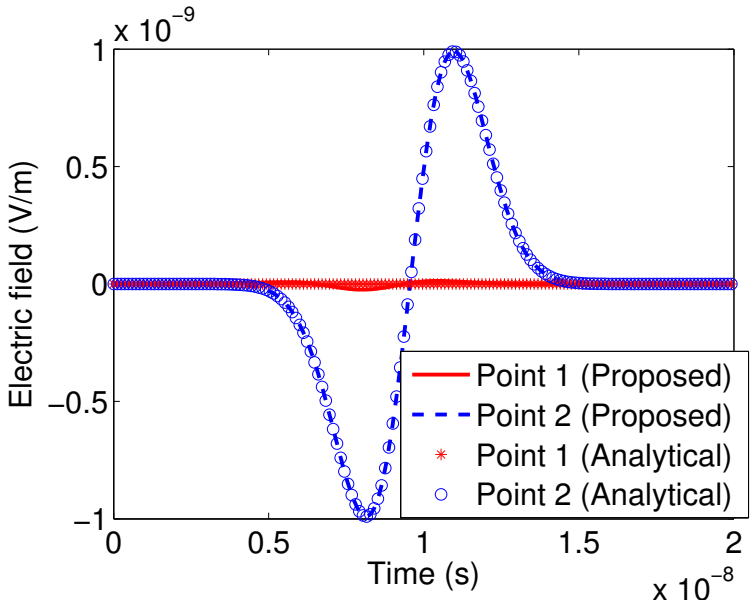



Fig. 5.4. Simulation of a 3D domain with a tetrahedral mesh: electric field at observation points.

as a capacitor of capacitance $C = 5.9027$ pF, thus the voltage can be calculated as $-\frac{\tau^2}{C}\exp\left(-(t-t_0)^2/\tau^2\right)$ V. On the other hand, only null space contributes to the solution, and all the unstable eigenmodes should be removed. This results in a much larger time step that is 0.01 s for the proposed unconditionally stable matrix-free time-domain method. Therefore, it only takes 30.7393 s to finish the simulation. In Fig. 5.6, the voltage simulated from the proposed method in comparison with analytical solution is plotted as a function of time. Obviously, the simulated result agrees very well with the reference result, validating the accuracy of the proposed method.

## 5.4 Conclusion

In this chapter, we develop an unconditionally stable matrix-free time-domain method for analyzing general electromagnetic problems discretized into arbitrarily shaped unstructured meshes. This method does not require the solution of a system

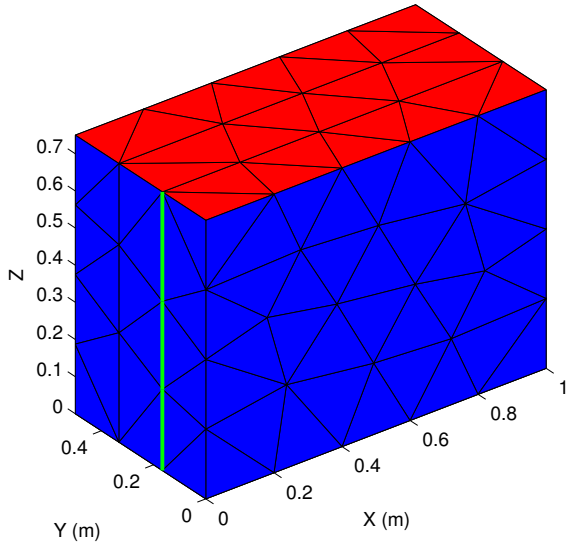



Fig. 5.5. Simulation of a parallel plate: Mesh details.

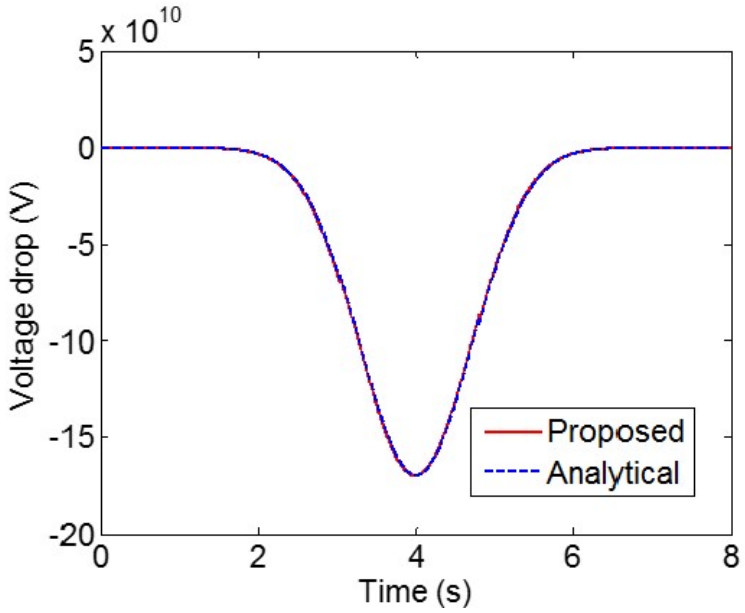



Fig. 5.6. Simulation of a parallel plate: Voltage drop between the two plates compared with analytical solution.

matrix, no matter which element shape is used for space discretization. Furthermore, this property is achieved irrespective of the time step used to perform the time domain

simulation. As a result, the time step can be solely determined by accuracy regardless of space step. Numerical experiments have validated the accuracy and efficiency of the proposed new method.

# 6. FAST EXPLICIT AND UNCONDITIONALLY STABLE FDTD METHOD

## 6.1 Introduction

Among so many time-domain methods, the finite-difference time-domain (FDTD) method is one of the most popular methods for electromagnetic analysis. This is mainly because of its simplicity and optimal computational complexity at each time step. However, as the matrix-free time-domain methods developed in previous chapters naturally reduce to the FDTD method in orthogonal grid, the time step of a conventional FDTD [1, 2] is also restricted by space step for stability, as dictated by the Courant-Friedrich-Levy (CFL) condition. If the space step of a given problem is determined solely from an accuracy point of view, the time step required by stability has a good correlation with the time step determined by accuracy. However, if the problem involves fine features relative to working wavelength, the time step required by stability can be orders of magnitude smaller than that required by accuracy. As a result, a large number of time steps must be simulated to finish one simulation, which is time consuming.

To overcome the aforementioned barrier, researchers have developed implicit unconditionally stable FDTD methods, such as the alternating-direction implicit (ADI) method [3, 4], the Crank-Nicolson (CN) method [5], the CN-based split step (SS) scheme [6], the pseudo-spectral time-domain (PSTD method) [7], the locally one-dimensional (LOD) FDTD [8, 9], the Laguerre FDTD method [10, 11], the associated Hermite (AH) type FDTD [12], a series of fundamental schemes [13] and many others, but the advantage of the conventional FDTD is sacrificed in avoiding a matrix solution. When the problem size is large, the implicit unconditionally stable FDTD methods become inefficient. Research has also been pursued to address the time

step problem in the original explicit time-domain methods [14–16]. In [17, 18], the source of instability is identified, and subsequently eradicated from the underlying numerical system to make an explicit FDTD unconditionally stable. It is shown that the source of instability is the eigenmodes of the discretized curl-curl operator whose eigenvalues are the largest. These eigenvalues are higher than what can be stably simulated by the given time step. To find these unstable modes, in [18], a partial solution of a global eigenvalue solution is computed. In general, only a small set of the largest eigenpairs of the system matrix need to be found, and the system matrix is also sparse. The same idea is also applied to the matrix-free time-domain method in Chap. 5 to solve time step problem. However, the computational overhead of the resultant scheme may still be too high to tolerate when the matrix size is large.

The time step required for a stable explicit simulation is limited by the largest eigenvalue of the system matrix. However, the finer the space step, the larger the largest eigenvalues of the system matrix. Therefore, there should exist a relationship between the fine cells present in a space discretization and the unstable modes that cannot be stably simulated by the given time step. We do not have to perform a brute-force eigenvalue solution to identify the unstable modes. Instead, we can utilize the relationship between the fine cells and the unstable modes to develop a more efficient explicit and unconditionally stable method. Along this line of thought, in this work, we first develop a new patch-based single-grid FDTD formulation. Using this formulation, we identify the theoretical relationship between fine cells and the largest eigenmodes of the underlying system matrix. We prove that once there exists a difference between the time step required by stability and the time step determined by accuracy, i.e., a difference between the fine-cell size and the regular-cell size, the largest eigenmodes of the original system matrix can be extracted from fine cells. The larger the contrast ratio between the two time steps, the more accurate the eigenmodes extracted in this way. Based on this theoretical finding, we propose an efficient algorithm to find the unstable modes directly from fine cells, and subsequently deduct these unstable modes from the numerical system to achieve an explicit time

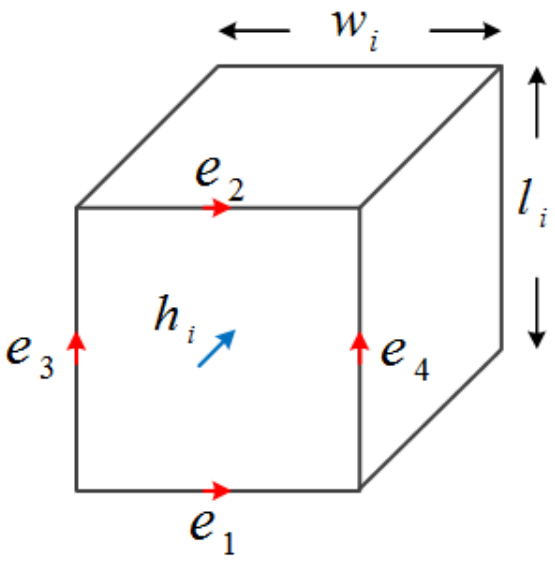



Fig. 6.1. Illustration of a patch-based discretization of Faraday's law.

marching with unconditional stability. The essential idea of this work can also be applied to other time-domain methods.

## 6.2 New Patch-Based Single-Grid FDTD Formulation

Before developing the proposed method, we first present a new formulation of the FDTD method, which is a patch-based single-grid formulation. Different from existing matrix-based FDTD formulations, this formulation reveals a natural decomposition of the curl-curl operator into a series of rank-1 matrices, which facilitates the development of the proposed method. The formulation does not require a dual grid, and it also shows each rank-1 matrix is positive semi-definite.

Consider a general 3-D grid. In every patch, from Faraday's law, the curl of **E** produces **H** as

$$\frac{e_2 - e_1}{l_i} + \frac{e_3 - e_4}{w_i} = -\mu \frac{\partial h_i}{\partial t}, \tag{6.1}$$

as illustrated in Fig. 6.1. The above can be rewritten as a row vector multiplied by a column vector

$$\left[-\frac{1}{l_i}, \frac{1}{l_i}, \frac{1}{w_i}, -\frac{1}{w_i}\right] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = -\mu \frac{\partial h_i}{\partial t}, \tag{6.2}$$

in which, $e$ denotes the tangential electric field at the center point of every edge in a patch, and $h$ denotes the magnetic field normal to the patch at the patch center. The $l_i$ and $w_i$ are, respectively, the two side lengths of the $i$-th patch, and $\mu$ is the permeability at the center of the $i$-th patch.

Define a global unknown vector $\{e\}$ that consists of all of the $e$ unknowns, and $\{h\}$ that contains all of the $h$ unknowns in the 3-D grid. We have

$$\{e\} = \{e_1, e_2, e_3, ..., e_{N_e}\}^T, \tag{6.3}$$

$$\{h\} = \{h_1, h_2, h_3, ..., h_{N_h}\}^T. \tag{6.4}$$

Clearly, the total number of **E** unknowns, $N_e$, is also the total number of edges in a 3-D grid. The total number of **H** unknowns, $N_h$, is the total number of patches. Using global vectors (6.3), (6.2) can be rewritten as

$$\mathbf{S}_e^{(ri)}{}_{1\times N_e}\{e\} = -\mu \frac{\partial \{h\}_i}{\partial t}, \tag{6.5}$$

where $\{h\}_i$ denotes the $i$-th entry of global vector $\{h\}$, and $\mathbf{S}_e^{(ri)}$ has the following expression

$$\mathbf{S}_e^{(ri)} = \left[-\frac{1}{l_i}, \frac{1}{l_i}, \frac{1}{w_i}, -\frac{1}{w_i}\right] \oplus zeros(1, N_e), \tag{6.6}$$

which is the row vector in (6.2) augmented with zeros to extend to length $N_e$, whose unknowns are ordered based on the global indexes of $e$-unknowns.

Writing (6.5) for each patch, and combing the resultant $N_h$ equations, we obtain the following matrix equation

$$(\mathbf{S}_e)_{N_h\times N_e}\{e\} = -diag\{\mu\}\frac{\partial \{h\}}{\partial t}, \tag{6.7}$$

which is essentially how Faraday's law is discretized in a FDTD grid. The $diag\{\mu\}$ denotes a diagonal matrix of permeability. It is evident that $\mathbf{S}_e$ is a sparse matrix of size $N_h \times N_e$, with each row being Faraday's law written for a single patch, and hence having 4 nonzero entries only.

The discretized Ampere's law in the FDTD method is nothing but the following matrix equation

$$(\mathbf{S}_h)_{N_e \times N_h} \{h\} = diag\{\epsilon\} \frac{\partial \{e\}}{\partial t} + \{j\}, \tag{6.8}$$

where $diag\{\epsilon\}$ is a diagonal matrix of permittivity, and $\{j\}$ denotes a current source vector. The $\mathbf{S}_h$ has a simple relationship with $\mathbf{S}_e$ as the following

$$\mathbf{S}_h = \mathbf{S}_e^T, \tag{6.9}$$

in a uniform grid. Hence, after (6.7) is obtained, (6.8) can be obtained immediately. In a non-uniform grid, the $\mathbf{S}_e$ stays the same; the $\mathbf{S}_h$ preserves the original sparse pattern, but the $l_i$ and $w_i$ are altered to use a length or width averaged between adjacent patches to yield a better accuracy. Specifically, the $l_i$ and $w_i$ are changed to the average size between the two patches sharing the same $\mathbf{E}$ unknown. Obviously, the aforementioned new approach for formulating the FDTD method is a single-grid, and patch-based approach. Its implementation is even more convenient than the original FDTD method.

The (6.7) and (6.8) can be combined to solve in a leap-frog way as the following

$$\{h\}^{n+\frac{1}{2}} = \{h\}^{n-\frac{1}{2}} - \Delta t \mathbf{D}_{\frac{1}{\mu}} \mathbf{S}_e \{e\}^n \tag{6.10}$$

$$\{e\}^{n+1} = \{e\}^n + \Delta t \mathbf{D}_{\frac{1}{\epsilon}} \mathbf{S}_h \{h\}^{n+\frac{1}{2}} - \Delta t \mathbf{D}_{\frac{1}{\epsilon}} \{j\}^{n+\frac{1}{2}}, \tag{6.11}$$

where superscripts $n$, $n+1$, and $n \pm \frac{1}{2}$ denote time instants, $\Delta t$ represents time step, $\mathbf{D}_{\frac{1}{\epsilon}}$ and $\mathbf{D}_{\frac{1}{\mu}}$ are diagonal matrices of $\frac{1}{\epsilon}$, and $\frac{1}{\mu}$ respectively.

The two first-order equations (6.7) and (6.8) can also be solved by eliminating $\{h\}$, obtaining a second-order equation in time for $\{e\}$ as the following

$$\frac{\{e\}^{n+1} - 2\{e\}^n + \{e\}^{n-1}}{\Delta t^2} + \mathbf{D}_{\frac{1}{\epsilon}} \mathbf{S}_h \mathbf{D}_{\frac{1}{\mu}} \mathbf{S}_e \{e\}^n = \{f\}^n, \tag{6.12}$$

where $\{f\}$ denotes the terms moved to the right hand side when deriving (6.12). The above is actually a central-difference based discretization of

$$\frac{\partial^2\{e\}}{\partial t^2} + \mathbf{S}\{e\} = \{f\}, \tag{6.13}$$

where

$$\mathbf{S} = \mathbf{D}_{\frac{1}{\epsilon}}\mathbf{S}_h\mathbf{D}_{\frac{1}{\mu}}\mathbf{S}_e, \tag{6.14}$$

which is a sparse matrix representing the discretized $\frac{1}{\epsilon}(\nabla\times)\frac{1}{\mu}(\nabla\times)$ operator.

In a conventional FDTD method, a matrix-less notation is used, which prevents one from seeing the structure of $\mathbf{S}$ easily. With the proposed formulation, from (6.6) and (6.9), it can be seen that $\mathbf{S}$ is the sum of $N_h$ rank-1 matrices as the following

$$\mathbf{S} = \mathbf{D}_{\frac{1}{\epsilon}}\sum_{i=1}^{N_h}\frac{1}{\mu_i}\mathbf{S}_h^{(ci)}{}_{N_e\times 1}\mathbf{S}_e^{(ri)}{}_{1\times N_e}. \tag{6.15}$$

Basically, we loop over all the patches in a 2- or 3-D grid. For each patch $i$ ($i = 1, 2, ..., N_h$), we generate a single column $\mathbf{S}_h^{(ci)}$, and a single row $\mathbf{S}_e^{(ri)}$. Multiplying the two together is the contribution of this patch to the entire $\mathbf{S}$, which is a rank-1 matrix, and also positive semi-definite as can be seen from (6.6) and (6.9). The $\mathbf{S}$ can then be obtained as the summation of the rank-1 matrix of each patch.

Because of (6.15), mathematically, it becomes possible to find its largest $k$ eigenvectors from its $k$ columns and $k$ rows having the largest norm. These columns and rows correspond to exactly those contributed by fine patches. To see this point more clearly, let the sequence of $\mathbf{S}_h^{(c1)}, \mathbf{S}_h^{(c2)}, ...$ be in a descending order of vector norm, with $\mathbf{S}_h^{(c1)}$'s norm being the largest, and $\mathbf{S}_h^{(c,k+1)}$'s norm $\epsilon$ times smaller than $\mathbf{S}_h^{(c1)}$'s norm. $\mathbf{S}$ can then be well approximated as $\tilde{\mathbf{S}} = \mathbf{D}_{\frac{1}{\epsilon}}\sum_{i=1}^{k}\frac{1}{\mu_i}\mathbf{S}_h^{(ci)}\mathbf{S}_e^{(ri)}$, with the error of $||\mathbf{S} - \tilde{\mathbf{S}}||/||\mathbf{S}||$ bounded by $O(\epsilon^2)$. Hence, $\tilde{\mathbf{S}}$ can be sufficient for finding $m \leq k$ largest eigenvalues and their corresponding eigenvectors with good accuracy, although it cannot be used to find all eigenpairs. The above analysis can still be conceptual. In the following section, we will provide a detailed proof.

## 6.3 Proposed Method for Lossless Problems

### 6.3.1 Theoretical Analysis

As shown in [17,18], in a conventional FDTD method, the time step for stability, $\Delta t_s$, is required to satisfy the following criterion

$$\Delta t_s \leq \frac{2}{\sqrt{\rho(\mathbf{S})}}, \tag{6.16}$$

where $\rho(\mathbf{S})$ denotes the spectral radius of $\mathbf{S}$, which is the largest eigenvalue of $\mathbf{S}$. Since this eigenvalue is inversely proportional to the smallest space step, (6.16) also dictates that the maximum time step permitted by stability depends on the smallest space step. In [17,18], the eigenvectors of $\mathbf{S}$ corresponding to the largest eigenvalues, which are beyond the stability criterion, are identified as the root cause of instability. The Arnoldi algorithm is then employed to find these unstable eigenvectors. For a sparse matrix of size $N_e$, to find its largest $k$ eigenpairs may take many more than $k$ Arnoldi steps, with the computational complexity being $O(k'^2 N)$, where $k' > k$. When $N$ is large, the computational overhead for obtaining a complete set of unstable modes in [18] could still be too high to tolerate.

Given a time step, define the fine cells as those cells whose size is smaller than that permitted by the CFL condition. From (6.15), it can be seen that the matrix $\mathbf{S}$ is the sum of many rank-1 matrices, each of which corresponds to one patch. From (6.6), we can also see that the smaller the patch, the larger the norm of its rank-1 matrix. Hence, the smallest patches contribute the largest norm in the assembled $\mathbf{S}$. It is then possible to find the largest eigenvalues and their eigenvectors of $\mathbf{S}$ from the submatrices assembled from fine cells only. This also indicates that the field distribution of unstable modes is actually localized in fine cells. Next, we give a quantitative proof on this point.

$\mathbf{S}$ can be split into the following two components

$$\mathbf{S} = \mathbf{S}_f + \mathbf{S}_c, \tag{6.17}$$

where $\mathbf{S}_f$ is assembled from fine cells, and $\mathbf{S}_c$ from the rest. Consider an eigenvector, $\mathbf{F}_{hi}$, of $\mathbf{S}_f$. It satisfies

$$\mathbf{S}_f\mathbf{F}_{hi} = \lambda_i\mathbf{F}_{hi}. \tag{6.18}$$

We need to prove it also satisfies the following:

$$\mathbf{S}\mathbf{F}_{hi} = \lambda_i\mathbf{F}_{hi}. \tag{6.19}$$

If so, then the eigenvectors obtained from the fine cells are also the eigenvectors of the entire problem domain.

*Proof:* To prove (6.19), we evaluate the accuracy of

$$\epsilon_{acc} = \frac{\|\mathbf{S}\mathbf{F}_{hi} - \lambda_i\mathbf{F}_{hi}\|}{\|\mathbf{S}\mathbf{F}_{hi}\|}. \tag{6.20}$$

Since $\mathbf{S}\mathbf{F}_{hi} = (\mathbf{S}_f + \mathbf{S}_c)\mathbf{F}_{hi}$, and $\mathbf{F}_{hi}$ satisfies (6.18), (6.20) yields

$$\epsilon_{acc} = \frac{\|\mathbf{S}_c\mathbf{F}_{hi}\|}{\|\mathbf{S}_c\mathbf{F}_{hi} + \lambda_i\mathbf{F}_{hi}\|}. \tag{6.21}$$

Since $\mathbf{S}_c$ is semi-positive definite, the above satisfies

$$\epsilon_{acc} \leq \frac{\|\mathbf{S}_c\|\|\mathbf{F}_{hi}\|}{\lambda_i\|\mathbf{F}_{hi}\|} = \frac{\|\mathbf{S}_c\|}{\lambda_i}. \tag{6.22}$$

Since $\mathbf{S}_c$ is Hermitian, its norm is also its spectral radius, i.e., the largest eigenvalue of $\mathbf{S}_c$. This number determines the maximum time step that can be used in the regular cells for a stable simulation, $\Delta t_c$. Similarly, the maximum $\lambda_i$ of $\mathbf{S}_f$ determines the time step $\Delta t_f$ that can be used in the fine cells for a stable simulation, which is also equal to the $\Delta t_s$ in (6.16) for the entire computational domain. As a result, from (6.22), we obtain

$$\epsilon_{acc} \leq \left(\frac{\Delta t_f}{\Delta t_c}\right)^2 = \left(\frac{\Delta t_s}{\Delta t}\right)^2. \tag{6.23}$$

The last equality in the above holds true because the ratio of $\Delta t_f$ to $\Delta t_c$ is also the ratio of time step required by stability $\Delta t_s$ to that determined by solution accuracy ($\Delta t$), assuming the regular-cell region is discretized based on accuracy.

From (6.23), it is evident that once $\Delta t_s$ is smaller than $\Delta t$, which is exactly the scenario when the time step issue must be solved, the unstable eigenmodes can be obtained from fine cells. Meanwhile, the larger the contrast ratio of regular-grid size over fine-grid size, the better the accuracy of the unstable eigenmodes extracted from fine cells. In addition, from (6.22), it can be seen among the eigenvalues $\lambda_i$ obtained from the fine cells, the larger the eigenvalue, the better the accuracy.

Based on the above finding, we develop an algorithm to find the unstable modes from fine cells only, and subsequently deduct these unstable modes from the numerical system for an explicit time marching with unconditional stability. The details of this algorithm are given in next section.

### 6.3.2 Proposed Algorithm

The proposed method includes three steps. First, we find unstable modes accurately from fine cells with controlled accuracy. Second, we upfront deduct the unstable modes from the system matrix, and perform explicit marching on the updated system matrix with absolute stability. Last, we add back the contribution of unstable modes if necessary.

#### Step I: Finding unstable modes accurately from fine cells

Given any desired time step $\Delta t$, the proposed method starts from categorizing the cells in the grid into two groups. One group $\mathbb{C}_c$ has a regular cell size and permits the use of the desired time step, while the other group $\mathbb{C}_f$ includes all the fine cells and the cells immediately adjacent to the fine cells. These cells require a smaller time step for a stable simulation. The cells in group $\mathbb{C}_f$ can be arbitrarily located in the grid. They do not have to be connected. Accordingly, $\mathbf{S}$ can be split as shown in (6.17), where $\mathbf{S}_f$ is $\mathbf{S}$ assembled from $\mathbb{C}_f$, and $\mathbf{S}_c$ is from $\mathbb{C}_c$. To identify $\mathbf{S}_f$, the new FDTD formulation presented in Section II provides a convenient and efficient approach. Based on (6.15), we obtain $\mathbf{S}_f$ by looping over all the patches in the fine-

cell region. For each patch, we obtain a rank-1 matrix $\mathbf{S}_h^{(ci)}\mathbf{S}_e^{(ri)}$. We then sum them up to obtain

$$\mathbf{S}_f = \mathbf{D}_{\frac{1}{\epsilon}} \sum_{i=1, i\in\mathbb{C}_f}^{k} \frac{1}{\mu_i} \mathbf{S}_h^{(ci)}{}_{N_e\times 1} \mathbf{S}_e^{(ri)}{}_{1\times N_e}, \tag{6.24}$$

in which $k$ is the number of patches in $\mathbb{C}_f$.

Let the $\mathbf{E}$ and $\mathbf{H}$ unknown number in $\mathbb{C}_f$ be $n$, and $k$ respectively. Obviously, $n < N_e$, and $k < N_h$. The $\mathbf{S}_h^{(ci)}$ in (6.24) is only nonzero in the rows corresponding to the fine-cell unknowns. Similarly, the $\mathbf{S}_e^{(ri)}$ is only nonzero in the columns associated with the fine-cell unknowns. The (6.24) hence can be rewritten as a small $n$ by $n$ matrix

$$\mathbf{S}_f^{(f)}{}_{n\times n} = \mathbf{A}_{n\times k}\mathbf{B}_{k\times n}^T, \tag{6.25}$$

where $\mathbf{A}$ stores all the $k$ columns of $\mathbf{S}_h^{(ci)}$, and $\mathbf{B}^T$ consists of all the rows of $\mathbf{S}_e^{(ri)}$ with zeros corresponding to the regular-cell unknowns removed. The material property has been taken into consideration in $\mathbf{A}$ and $\mathbf{B}$. Since $k$ is less than $n$, the $\mathbf{S}_f^{(f)}$ is further a low-rank matrix. We then extract $l$ unstable eigenmodes $\mathbf{F}_{hi}$ ($l < k$, $i = 1, 2, \ldots, l$) from it, the complexity of doing so is only $O(l^2 n)$. This is much smaller than $O(k'^2 N_e)$ in [18], which is the complexity of a global eigenvalue solution with $k'$ Arnoldi steps for finding $k$ largest eigenpairs, since $l < k'$, and $n << N_e$. Basically, we find the largest $l$ eigenvalues $\lambda_i$ and their corresponding eigenvectors $\mathbf{F}_{hi}^{(f)}$ of the small $n \times n$ matrix $\mathbf{S}_f$ by using Arnorldi method. Given a threshold $\epsilon$, if the following requirement is satisfied, the $\mathbf{F}_{hi}^{(f)}$ is accurate enough to be included in the unstable modes,

$$\epsilon_{acc} = \frac{\|\mathbf{S}\mathbf{F}_{hi} - \lambda_i \mathbf{F}_{hi}\|}{\|\mathbf{S}\mathbf{F}_{hi}\|} < \epsilon, \tag{6.26}$$

where $\mathbf{F}_{hi}$ is $\mathbf{F}_{hi}^{(f)}$ extended to length $N_e$ based on the global unknown ordering. Among $l$ eigenvectors, assume $k_r$ of them are accurate. They are also corresponding to the $k_r$ largest eigenvalues. We then orthogonalize them as $\mathbf{V}_h$ for the use of next step.

When calculating $\epsilon_{acc}$, 2-norm is used in this work. The choice of the accuracy threshold $\epsilon$ is a user-defined parameter. Since the larger the eigenvalue, the better

the accuracy of the eigenmode extracted from $\mathbf{S}_f$, as shown in previous section, we compute the eigenvalues of $\mathbf{S}_f$ starting from the largest to smaller ones. For each eigenpair computed, we calculate $\epsilon_{acc}$ defined in (6.20) until it is greater than prescribed $\epsilon$. The $\epsilon_{acc}$ calculated for the largest eigenpair represents the best accuracy one can achieve in the given grid, which also dictates the smallest $\epsilon$ one can choose.

**Step II: Explicit and unconditionally stable time marching**

After the unstable modes are found, to make the explicit FDTD stable for the desired time step, we upfront deduct the contribution of $\mathbf{V}_h$ from $\mathbf{S}$ as follows

$$\mathbf{S}_l = \mathbf{S} - \mathbf{V}_h \mathbf{V}_h^H \mathbf{S}, \tag{6.27}$$

which allows for a much larger time step than $\mathbf{S}$. We then perform an explicit marching on the updated system matrix as

$$\{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} - \Delta t^2 \mathbf{S}_l \{e\}^n + \Delta t^2 \{f\}^n \tag{6.28}$$

followed by the following treatment to ensure the resultant $\{e\}$ has no component in $\mathbf{V}_h$ space

$$\{e\}^{n+1} = \{e\}^{n+1} - \mathbf{V}_h \mathbf{V}_h^H \{e\}^{n+1}. \tag{6.29}$$

Since the contribution of $\mathbf{V}_h$ is removed from $\mathbf{S}$, the time marching of (6.28) is stable for the desired large time step. In the extreme case where all cells are fine cells not allowing for the desired time step, the $\mathbf{S}_f$ becomes $\mathbf{S}$. Hence, only null space of $\mathbf{S}$ is left in $(\mathbf{S} - \mathbf{V}_h \mathbf{V}_h^H \mathbf{S})$, permitting an infinitely large time step.

**Step III: Adding back the contribution of unstable modes if necessary**

This step is not needed when the time step is chosen based on accuracy, since the unstable modes removed are not required by accuracy as analyzed in [18]. In the case when time step chosen is larger than that required by accuracy, some eigenvectors that are important to the field solution are also removed from the numerical system,

therefore the solution computed from (6.28) and (6.29) is no longer an accurate solution of the original problem in (6.13) any more. In this case, the proposed algorithm allows users to add the $\mathbf{V}_h$-contribution back to guarantee accuracy. Basically, the field solution $\{e\}$ of (6.13) can be expressed as

$$\{e\} = \mathbf{V}\{y\} = \mathbf{V}_l\{y_l\} + \mathbf{V}_h\{y_h\} = \{e_l\} + \{e_h\}, \tag{6.30}$$

where $\mathbf{V} = [\mathbf{V}_l, \mathbf{V}_h]$ is an orthogonal matrix of full rank $N_e$. Since $\{e_l\}$ has been obtained from (6.28) and (6.29), we only need to find $\{e_h\} = \mathbf{V}_h\{y_h\}$. Since $\mathbf{V}_h$ has been found, $\{y_h\}$ can be readily evaluated by front multiplying $\mathbf{V}_h^H$ on both sides of (6.13) to obtain

$$\frac{\partial^2\{y_h\}}{\partial t^2} + \mathbf{S}_r\{y_h\} = \mathbf{V}_h^T(\{f\} - \mathbf{S}\{e_l\}), \tag{6.31}$$

where $\mathbf{S}_r = \mathbf{V}_h^H\mathbf{S}\mathbf{V}_h$, whose size is the number of unstable modes $k_r$. The above can be solved efficiently by the method in [18]. Since the size is small, it can also be solved by implicit methods.

### 6.3.3 How It Works?

Apparently, since the proposed algorithm also allows one to add the $\mathbf{V}_h$ contribution back, it seems that any orthogonal space $\mathbf{V}_h$ would work. This is not true. For (6.28) and (6.29) to produce a correct solution, $\mathbf{V}_h$ needs to satisfy the property of $\mathbf{V}_l^T\mathbf{S}\mathbf{V}_h = 0$. To see this point clearly, we can start from (6.13). Since $\mathbf{S}$ has both $\mathbf{V}_h$ and $\mathbf{V}_l$ components, the solution $\{e\}$ also has both components. Thus, (6.13) can be rewritten as

$$\frac{\partial^2\left(\mathbf{V}_l\{y_l\} + \mathbf{V}_h\{y_h\}\right)}{\partial t^2} + \mathbf{S}\left(\mathbf{V}_l\{y_l\} + \mathbf{V}_h\{y_h\}\right) = \{f\}. \tag{6.32}$$

To obtain the $\mathbf{V}_l$-component of $\{e\}$, we can multiply the above by $\mathbf{V}_l^H$ from both sides. This yields

$$\frac{\partial^2\{y_l\}}{\partial t^2} + \mathbf{V}_l^H\mathbf{S}(\mathbf{V}_l\{y_l\} + \mathbf{V}_h\{y_h\}) = \mathbf{V}_l^H\{f\}. \tag{6.33}$$

If $\mathbf{V}_l^H \mathbf{S} \mathbf{V}_h$ does not vanish, (6.33) cannot be reduced to a numerical system of $\{y_l\}$ only. Only when $\mathbf{V}_l^H \mathbf{S} \mathbf{V}_h = 0$, front multiplying (6.33) by $\mathbf{V}_l$, we can obtain (6.28), where $\{e\} = \{e_l\}$ due to (6.29), and $\mathbf{I} - \mathbf{V}_h \mathbf{V}_h^H = \mathbf{V}_l \mathbf{V}_l^H$.

Since (6.26) is satisfied, $\mathbf{F}_h$ is an accurate eigenvector of $\mathbf{S}$. With $\mathbf{V}_h$ orthogonalized from $\mathbf{F}_h$, the property of $\mathbf{V}_l^H \mathbf{S} \mathbf{V}_h = 0$ is satisfied. This is because $\mathbf{S}\mathbf{V}_h = \mathbf{S}\mathbf{F}_h\mathbf{Z} = \mathbf{F}_h\mathbf{\Lambda}_h\mathbf{Z} = \mathbf{V}_h\mathbf{Z}^{-1}\mathbf{\Lambda}_h\mathbf{Z}$, and $\mathbf{V}_l^H \mathbf{V}_h = 0$. Here, we use the relationship of $\mathbf{V}_h = \mathbf{F}_h\mathbf{Z}$ where $\mathbf{Z}$ is a full-rank transformation matrix, as $\mathbf{V}_h$ is orthogonalized from $\mathbf{F}_h$.

### 6.3.4 Computational Efficiency

In the proposed method, we avoid finding the eigensolutions of the original global system matrix $\mathbf{S}$. Instead, we work on a much smaller matrix $\mathbf{S}_f$. Therefore, compared with the approach developed in [18], the proposed method can achieve unconditional stability more efficiently without sacrificing accuracy. The complexity of finding unstable modes is reduced significantly from the original $O(k'^2 N_e)$ to $O(l^2 n)$ with $n << N_e$, and $l < k'$. This small cost is also a one-time cost, which is performed before time marching. Since the unstable modes found in this work are frequency and time independent, once found, they can be reused for different simulations of the same physical structure. In the second step of explicit time-marching, the matrix-free property of the FDTD is preserved. The time marching has a strict linear (optimal) complexity at each time step.

## 6.4 Proposed Method for Lossy Problems

In previous section, we focus on lossless problems. When there exist lossy dielectrics and conductors, we need to add one more term to (6.13) as follows

$$\frac{\partial^2 \{e\}}{\partial t^2} + \mathbf{D}\frac{\partial \{e\}}{\partial t} + \mathbf{S}\{e\} = \{f\}, \tag{6.34}$$

where $\mathbf{D}$ is diagonal with its $i$-th entry being $\sigma_i/\epsilon_i$ at the point of the $i$-th $\mathbf{E}$ unknown. Different from a lossless problem, (6.34) is governed by the following quadratic eigenvalue problem

$$(\lambda^2 + \lambda\mathbf{D} + \mathbf{S})v = 0. \tag{6.35}$$

The treatment of such a problem is different from that of a generalized eigenvalue problem. We hence use a separate section to describe our solution to general lossy problems.

### 6.4.1 Theoretical Analysis

The second-order differential equation (6.34) can be transformed to the following first-order equation in time without any approximation

$$\frac{\partial\{\tilde{e}\}}{\partial t} - \mathbf{M}\{\tilde{e}\} = \{\tilde{f}\}, \tag{6.36}$$

where $\{\tilde{f}\} = [0 \quad f]^T$, $\{\tilde{e}\} = [e \quad \dot{e}]^T$, in which $\dot{e}$ denotes the first-order time derivative of $e$, and matrix $\mathbf{M}$ is

$$\mathbf{M} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{S} & -\mathbf{D} \end{bmatrix}, \tag{6.37}$$

where $\mathbf{I}$ is an identity matrix. Obviously, $\{\tilde{e}\}$'s upper part is the original field solution of (6.34).

The solution of (6.36) is governed by the following generalized eigenvalue problem

$$\mathbf{M}x = \lambda x. \tag{6.38}$$

This problem is also equivalent to (6.35) by using the relationship of $x = [v \quad \lambda v]^T$. Since $\mathbf{I}$ is positive definite, $\mathbf{D}$ is semi-positive definite, and $\mathbf{S}$ is semi-positive definite, the eigenvalues of (6.38) either are non-positive real or come as complex conjugate pairs whose real part is less than zero. Similar to lossless problems, to achieve unconditional stability, we also need to remove the unstable modes from the system matrix,

now $\mathbf{M}$. These modes are analyzed in [55]. They have eigenvalues whose magnitude satisfies

$$|\lambda| > \frac{2}{\Delta t}. \tag{6.39}$$

Again, given a desired time step, the unstable modes have the largest eigenvalues in magnitude. Compared to lossless problems, now it is even more computationally expensive to find these unstable modes since $\mathbf{M}$ is double sized and can be highly ill-conditioned when conductor loss is involved. Therefore, similar to what we do for lossless problems, we propose to find the unstable modes efficiently from the fine cells only.

### 6.4.2 Proposed Method

When dealing with lossless problems, all the cells in the computational domain are divided into two groups, $\mathbb{C}_f$ and $\mathbb{C}_c$, based on the time step permitted by their grid size. For lossy problems, we incorporate into $\mathbb{C}_f$ not only the fine cells and their immediately adjacent cells, but also all the cells filled with conductive metals. This is because the conductive materials contribute eigenvalues as large as conductivity divided by permittivity. To explain, the lowest eigenmode of (6.35) satisfies $\mathbf{S}v = 0$, which is a gradient field. For this field, in addition to zero eigenvalues, there is a set of eigenvalues whose magnitude is approximately $||\mathbf{D}||$, which is $\sigma$ over permittivity. Hence, the conductive region is included since unstable modes correspond to the largest eigenvalues.

After $\mathbb{C}_f$ is identified, we can form a matrix $\mathbf{M}_f$ as follows

$$\mathbf{M}_f = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{S}_f & -\mathbf{D}_f \end{bmatrix}, \tag{6.40}$$

where $\mathbf{S}_f$ can be found in the same way as (6.25), $\mathbf{D}_f$ is obtained by selecting the diagonal entries of $\mathbf{D}$ corresponding to the field unknowns in $\mathbb{C}_f$. As a result, $\mathbf{M}_f$ is a $2n \times 2n$ matrix, which is much smaller than the original size of $\mathbf{M}$. We then extract the largest eigenpairs of $\mathbf{M}_f$ by using the Arnorldi method. Similarly, an accuracy check

similar to (6.26) (with $\mathbf{S}$ replaced by $\mathbf{M}$) is performed to select accurate unstable modes obtained from $\mathbf{M}_f$. Let $k_r$ be the unstable eigenmodes obtained from $\mathbf{M}_f$, the complexity of finding them is simply $O(k_r^2 n)$. We then orthogonalize the unstable modes obtained, and augment them with zeros based on the global unknown indexes to build $\mathbf{V}_h$.

Using $\mathbf{V}_h$, we upfront deduct their contributions from the system matrix before time marching as follows:

$$\mathbf{M}_l = \mathbf{M} - \mathbf{V}_h \mathbf{V}_h^H \mathbf{M}. \tag{6.41}$$

We then perform a time marching of (6.36) using the updated system matrix $\mathbf{M}_l$ as the following:

$$\frac{\partial\{\tilde{e}\}}{\partial t} - \mathbf{M}_l\{\tilde{e}\} = \{\tilde{f}\}. \tag{6.42}$$

If we perform a forward-difference-based time marching on (6.42), the resultant update equation is definitely explicit. However, the stability requirement on the time step is $\Delta t \leq -2Re(\lambda)/|\lambda|^2$ where $\lambda$ is the eigenvalue of $\mathbf{M}_l$. This results in a time step smaller than $\Delta t \leq 2/|\lambda|$, which is the time step required by a central-difference discretization of the original second-order equation (6.34), for stably simulating the same set of $\lambda$. To solve this problem, we propose to perform a backward difference as shown below

$$(\mathbf{I} - \Delta t \mathbf{M}_l)\{\tilde{e}\}^{n+1} = \{\tilde{e}\}^n + \Delta t\{\tilde{f}\}^{n+1}. \tag{6.43}$$

A $z$-transform of the above results in $z = 1/(1 - \lambda\Delta t)$. Since $\lambda$ of $\mathbf{M}_l$ has a non-positive real part, the stability of (6.43) is ensured for any large time step. Using the accuracy determined time step $\Delta t$, and with the corresponding unstable modes removed, all the eigenvalues of $\mathbf{M}_l$ satisfy

$$|\lambda| \leq \frac{1}{\Delta t}. \tag{6.44}$$

Hence, the inversion of the left hand matrix of (6.43) can be replaced by a series expansion with a small number of terms. Thus, (6.43) can be explicitly marched on in time as the following

$$\{\tilde{e}\}^{n+1} \approx \left(\mathbf{I} + \Delta t\mathbf{M}_l + (\Delta t\mathbf{M}_l)^2 + \ldots + (\Delta t\mathbf{M}_l)^p\right)\{\tilde{y}\}, \tag{6.45}$$

where $\{\tilde{y}\}$ represents the right-hand-side term in (6.43). In the above, there is no need to compute the matrix-matrix product. Instead, (6.45) is a summation of $p$ vectors, and every vector can be obtained by multiplying the previous vector by $\mathbf{M}_l$. Hence, the computational cost of (6.45) is simply $p$ matrix-vector multiplications, and $p < 10$.

To make sure the solution is free of unstable modes, we need to add the following treatment after (6.45) at each time instant

$$\{\tilde{e}\}^{n+1} = \{\tilde{e}\}^{n+1} - \mathbf{V}_h \mathbf{V}_h^H \{\tilde{e}\}^{n+1}. \tag{6.46}$$

### 6.4.3 Matrix Scaling

When conductor loss and/or multiscale structures are involved, $\mathbf{I}$, $\mathbf{D}$, and $\mathbf{S}$ can be orders of magnitude different in their matrix norm. The solution of the generalized eigenvalue problem (6.38) may have a poor accuracy. To improve the accuracy of finding unstable modes from $\mathbf{M}_f$, we adopt an optimal scaling technique introduced in [56]. Based on this technique, the $\mathbf{I}$ and $\mathbf{S}$ in (6.37) are scaled to

$$\tilde{\mathbf{I}} = \alpha \mathbf{I}, \quad \tilde{\mathbf{S}} = \mathbf{S}/\alpha, \tag{6.47}$$

where

$$\alpha = \sqrt{\|\mathbf{S}\|_2}. \tag{6.48}$$

Consequently, the first-order double-sized system (6.36) is updated as follows

$$\frac{\partial\{\tilde{e}'\}}{\partial t} - \tilde{\mathbf{M}}\{\tilde{e}'\} = \{\tilde{f}'\}, \tag{6.49}$$

where $\{\tilde{e}'\} = [e \quad \dot{e}/\alpha]^T$, $\{\tilde{f}'\} = [0 \quad f/\alpha]^T$, and $\tilde{\mathbf{M}}$ is

$$\tilde{\mathbf{M}} = \begin{bmatrix} 0 & \tilde{\mathbf{I}} \\ -\tilde{\mathbf{S}} & -\mathbf{D} \end{bmatrix}. \tag{6.50}$$

The $\mathbf{M}_f$ formulated for fine cells is also scaled accordingly. As can be seen in (6.49), the upper half of the solution vector $\{\tilde{e}'\}$ is the same as that of (6.36).

## 6.5 Numerical Results

In this section, we simulate a number of 2- and 3-D examples involving inhomogeneous materials and lossy conductors to demonstrate the validity and efficiency of the proposed fast unconditionally stable FDTD method.

### 6.5.1 2-D Wave Propagation and Cavity Problems

We first simulate a wave propagation problem in a 2-D rectangular region. The grid is shown in Fig. 6.2, where fine cells are introduced to examine the unconditional stability of the proposed method. Along $y$-axis, the cell size is uniform of 0.1 m. Along $x$-axis, we define $Contrast\ Ratio = \Delta x_c/\Delta x_f$ where $\Delta x_c = 0.1$ m, and $\Delta x_f$ is controlled by *Contrast Ratio*. There are three fine cells along $x$ axis whose cell size is $\Delta x_f$. The total number of $\mathbf{E}$ unknowns is 258. The incident electric field is $\mathbf{E}^{inc} = \hat{y}2(t - t_0 - x/c)e^{-(t-t_0-x/c)^2/\tau^2}$ with $c = 3 \times 10^8$ m/s, $\tau = 2 \times 10^{-8}$ s and $t_0 = 4\tau$. The regular grid size, $\Delta x_c = 0.1$ m, satisfies accuracy for capturing frequencies present in the input spectrum, which is about 1/20 of the smallest wavelength. The computational domain is terminated by an exact absorbing boundary condition, which is the known total field. This is because for any problem, the total fields on the boundary serve as an exact absorbing boundary condition to truncate a computational domain. For most of the problems, such fields are unknown. However, in a free-space problem studied in this example, the total field is known since it is equal to the incident field.

When choosing $Contrast\ Ratio = 100$, $\Delta x_f = 0.001$ m, which is two orders of magnitude smaller than that required by accuracy. Hence, there is a two orders of magnitude difference between the time step required by accuracy and that by stability. The conventional FDTD method must use a time step no greater than $3.84 \times 10^{-12}$ s to perform a stable simulation. In contrast, the proposed method is able to use a time step of $2.42 \times 10^{-10}$ s solely determined by accuracy to carry out the simulation. The fine patches and their adjacent patches are identified, which are marked in red

Table 6.1
The largest 16 eigenvalues obtained from $\mathbf{S}_f$ when $Contrast\ Ratio = 100$

| | |
|---|---|
| $\lambda_1$ | 2.70260697E+25 |
| $\lambda_2$ | 2.70251709E+25 |
| $\lambda_3$ | 2.70240599E+25 |
| $\lambda_4$ | 2.70228179E+25 |
| $\lambda_5$ | 2.70231612E+25 |
| $\lambda_6$ | 9.16900389E+24 |
| $\lambda_7$ | 9.16810512E+24 |
| $\lambda_8$ | 9.16699417E+24 |
| $\lambda_9$ | 9.16575210E+24 |
| $\lambda_{10}$ | 9.166095405E+24 |
| $\lambda_{11}$ | 1.23429421E+22 |
| $\lambda_{12}$ | 1.22530647E+22 |
| $\lambda_{13}$ | 1.21419701E+22 |
| $\lambda_{14}$ | 1.20177625E+22 |
| $\lambda_{15}$ | 1.20520926E+22 |
| $\lambda_{16}$ | 5.89575274E+19 |

in Fig. 6.2. They involve 50 internal $\mathbf{E}$ unknowns. Therefore, the size of $\mathbf{S}_f$ is 50 by 50, from which 15 unstable eigenmodes are found accurately for a prescribed accuracy of $\epsilon = 10^{-6}$. The $\epsilon_{acc}$ for the 16-th eigenmode in (6.26) is 0.1036. Hence, the 16-th eigenmode and thereafter are not selected since their accuracy does not meet requirements. The 15 unstable modes are then deducted from the system matrix, permitting a two-orders-of-magnitude larger time step. In Fig. 6.3(a), the electric fields at two observation points marked by blue cross in Fig. 6.2 are plotted as a function of time. Obviously, they agree very well with reference analytical solutions.

In this example, we have also numerically examined whether the eigenmodes extracted from the fine cells are accurate approximations of the eigenmodes of the entire problem. In Table 6.1, we list the eigenvalues of the 15 unstable modes and also the 16-th one we extract from $\mathbf{S}_f$ with *Contrast Ratio* = 100. It is clear to see that the largest 15 eigenvalues are at least two orders of magnitude larger than the 16-th one. Once they are removed, a much larger time step can be used for a stable simulation. In Table 6.2, we list the accuracy of each unstable eigenmode with respect to different *Contrast Ratio* from 2, 5, 10, to 100, by calculating the relative error shown in (6.20). Obviously, for all these contrast ratios, the eigenmodes extracted from fine cells are shown to be accurate eigenmodes of the entire $\mathbf{S}$. Furthermore, the larger the contrast ratio between fine cells and coarse ones, the better the accuracy of the eigenmodes found from fine cells. Moreover, the eigenmodes whose eigenvalues are larger are more accurate. All of these have verified our theoretical analysis given in Section 6.3. Notice that when *Contrast Ratio* = 2, the number of unstable eigenmodes that can be accurately extracted is smaller. However, we still can obtain a set of eigenmodes accurately for such a small contrast ratio.

To examine the solution accuracy in the entire computational domain, we define the entire solution error at each time instant as $\|\{e\} - \{e\}_{anal}\|/\|\{e\}_{anal}\|$, where $\{e\}$ consists of *all* $\mathbf{E}$ unknowns simulated from the proposed method and $\{e\}_{anal}$ is the analytical solution to all the unknowns. For example, consider an $\mathbf{E}$ unknown located at $\mathbf{r}_i$ with direction $\hat{t}_i$, its analytical solution for this wave propagation problem is simply $\mathbf{E}^{inc}(\mathbf{r}_i)\cdot\hat{t}_i$. Two-norm is used to calculate the entire solution error. Meanwhile, we examine the solution accuracy as a function of *Contrast Ratio*. The entire solution error is plotted in Fig. 6.3(b) for four different *Contrast Ratio* 2, 5, 10 and 100 respectively. It is evident that the solution accuracy of the proposed method is satisfactory for all these contrast ratios. Furthermore, the larger the contrast ratio, the better the accuracy. It is known that a discretization with a high contrast ratio may yield inaccurate solutions. However, it is not the case in this example, since the reference solution used to plot the error in Fig. 6.3(b) is *analytical* solution, and the

error is shown to be less than one percent for all contrast ratios examined. To further examine this point, we also simulate the same problem using the conventional FDTD method with $\Delta t = 3.84 \times 10^{-12}$ s when *Contrast Ratio* = 100, and plot the entire solution error versus time in Fig. 6.4. As can be seen, even though the contrast ratio is large, the accuracy of the conventional FDTD method is still very good in this example. In addition, comparing Fig. 6.3(b) with Fig. 6.4 for *Contrast Ratio* = 100, it is obvious that the proposed method can achieve the same level of accuracy as the conventional FDTD method. As for efficiency, the CPU time speedup is 1.58, 3.08 and 28.16 respectively for contrast ratio being 5, 10 and 100. However, no speedup is observed when contrast ratio is 2, because of the small time step difference and the additional overhead of the proposed method. The proposed method takes 0.0563 s including the CPU time of every step from finding the unstable eigenmodes to explicit time marching, while the conventional FDTD method only requires 0.0367 s to finish the simulation.

When *Contrast Ratio* = 100, we also study a cavity problem in the same mesh shown in Fig. 6.2. All the boundary unknowns are truncated by a perfectly electric wall. A current source is placed at (0.4, 0.25) m, and its derivative is $\frac{\partial j}{\partial t} = 2(t - t_0)\exp^{-(t-t_0)^2/\tau^2}$ with $\tau = 2.0\times10^{-9}$ s and $t_0 = 4\tau$. The fine cells identified to assemble $\mathbf{S}_f$ are the same as those in the previous wave propagation problem. After the 15 unstable eigenmodes obtained from $\mathbf{S}_f$ are removed from system matrix, the proposed method can use a time step $2.4 \times 10^{-10}$ s while the conventional FDTD method is only allowed to use $\Delta t = 3.84 \times 10^{-12}$ s. In Fig. 6.5, the electric field sampled at point (0.4, 0.35) m is plotted. The reference solution is obtained by simulating the same problem using the conventional FDTD method. Again, the solution solved by the proposed method matches very well with the reference solution.

In Fig. 6.6, we also plot three eigenvectors of $\mathbf{S}$ whose eigenvalues are respectively the largest, the 5th largest, and the 15-th largest eigenvalues of global $\mathbf{S}$, for a contrast ratio of 100. As can be seen from Fig. 6.6, the field distributions of these eigenvectors are localized in the fine-cell region, with the fields in the regular cells many orders of

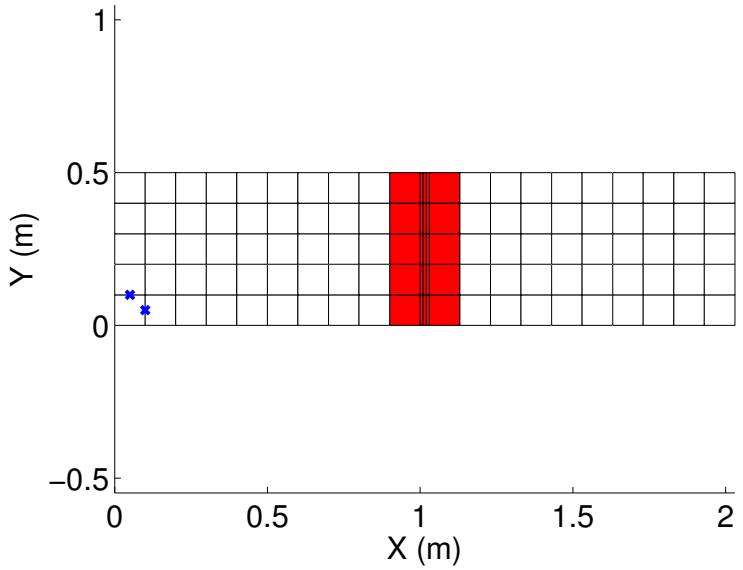
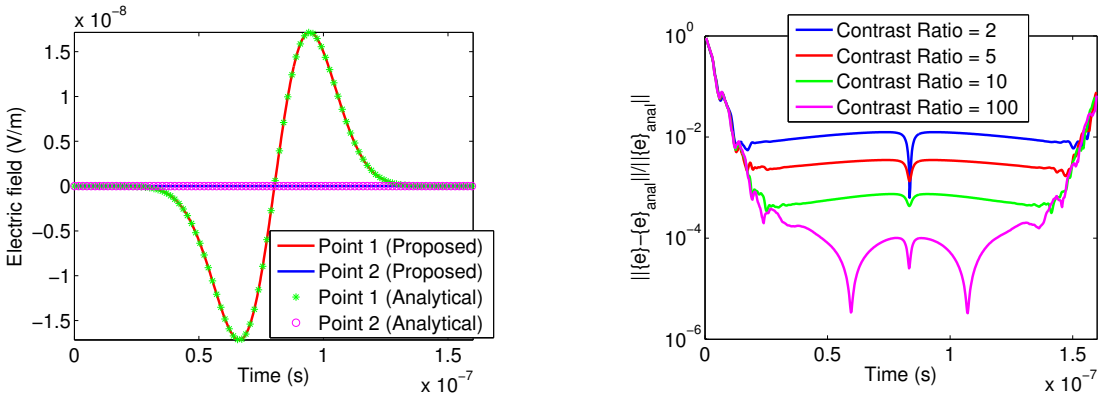



Fig. 6.2. Wave propagation in a 2-D rectangular region: Space discretization.




(a) (b)

Fig. 6.3. Wave propagation in a 2-D rectangular region: (a) Waveform of electric fields at two observation points when *Contrast Ratio* = 100. (b) *Entire* solution error v.s. time with different *Contrast Ratio* from 2, 5, 10 to 100.

magnitude smaller. For example, for the 15th largest eigenmode whose field distribution is more spread over than the first two, its eigenmode (eigenvector) still has a field value in the immediately adjacent coarse cells being three orders of magnitude

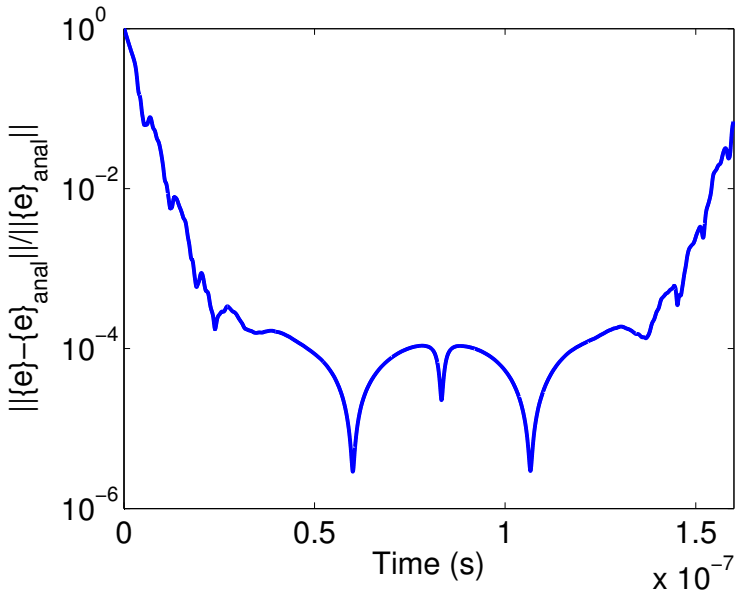



Fig. 6.4. Conventional FDTD for wave propagation problem: *Entire* solution error v.s. time with *Contrast Ratio* = 100.

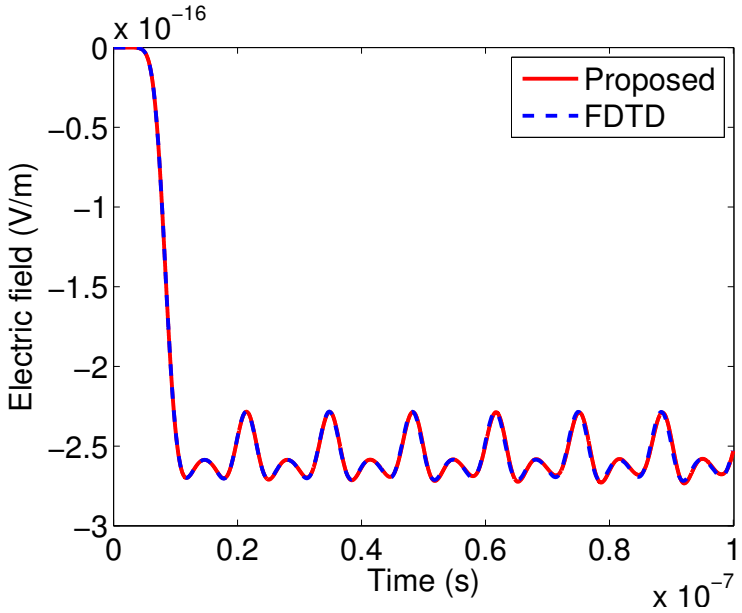



Fig. 6.5. Cavity problem: Waveform of electric fields at two observation points when *Contrast Ratio* = 100.

Table 6.2
The accuracy of each unstable eigenmode obtained from $\mathbf{S}_f$ with different contrast ratios

| $CR$ | 2 | 5 | 10 | 100 |
|---|---|---|---|---|
| $\mathbf{F}_{h1}$ | 2.9e-3 | 4.4e-5 | 1.6e-6 | 1.7e-11 |
| $\mathbf{F}_{h2}$ | 2.9e-3 | 4.1e-5 | 1.4e-6 | 1.6e-11 |
| $\mathbf{F}_{h3}$ | 2.8e-3 | 3.6e-5 | 1.2e-6 | 1.4e-11 |
| $\mathbf{F}_{h4}$ | 2.6e-3 | 3.0e-5 | 1.0e-6 | 1.1e-11 |
| $\mathbf{F}_{h5}$ | 2.5e-3 | 2.8e-5 | 9.6e-7 | 1.0e-11 |
| $\mathbf{F}_{h6}$ | 1.7e-2 | 6.0e-4 | 3.0e-5 | 4.5e-10 |
| $\mathbf{F}_{h7}$ | 1.8e-2 | 5.7e-4 | 2.7e-5 | 4.0e-10 |
| $\mathbf{F}_{h8}$ | 1.9e-2 | 5.1e-4 | 2.4e-5 | 3.5e-10 |
| $\mathbf{F}_{h9}$ | 1.9e-2 | 4.5e-4 | 2.0e-5 | 3.0e-10 |
| $\mathbf{F}_{h10}$ | 1.9e-2 | 4.2e-4 | 1.9e-5 | 2.7e-10 |
| $\mathbf{F}_{h11}$ | | 2.2e-2 | 6.8e-3 | 8.3e-5 |
| $\mathbf{F}_{h12}$ | | 2.4e-2 | 7.1e-3 | 8.2e-5 |
| $\mathbf{F}_{h13}$ | | 2.7e-2 | 7.4e-3 | 8.0e-5 |
| $\mathbf{F}_{h14}$ | | 2.8e-2 | 7.6e-3 | 7.7e-5 |
| $\mathbf{F}_{h15}$ | | 2.9e-2 | 7.5e-3 | 7.6e-5 |

smaller than that in the fine cells. This figure further confirms that the highest eigenmodes can be accurately extracted from fine cells. Although it is plotted for contrast ratio 100, similar localizations have been observed for other smaller contrast ratio, which can also be seen from the small error of eigenvectors extracted from $\mathbf{S}_f$ listed in Table 6.2. Numerically, such a localization is because the rapid field variation of the large-eigenvalue modes cannot be captured by a coarse discretization. This is similar to the fact that if one uses a coarse grid to extract the cavity resonance frequencies, the frequencies (eigenvalues) one can numerically identify are much smaller than the ones he can find when using a fine grid. Analytically, all these eigenvalues should

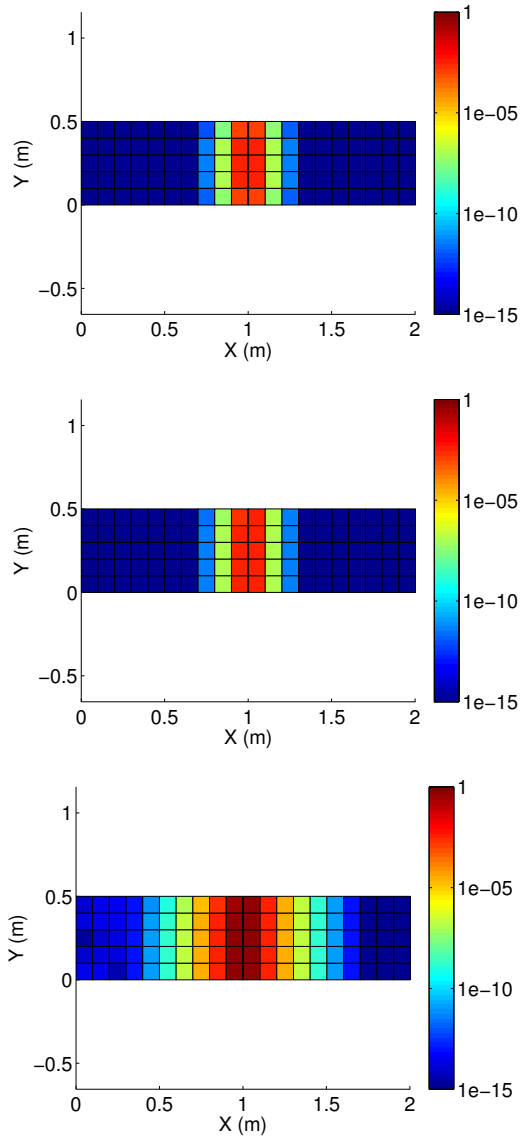
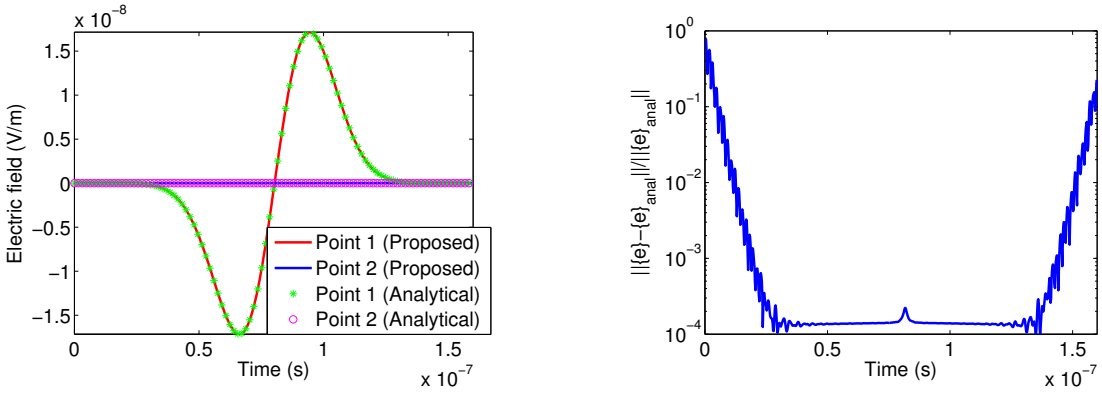



Fig. 6.6. Field distribution of the eigenvectors of **S** for a contrast ratio of 100 plotted in log scale: (a) Eigenvector having the largest eigenvalue. (b) Eigenvector having the 5th largest eigenvalue. (c) Eigenvector having the 15th-largest eigenvalue.

exist in the solution domain. However, numerically, only finer cells can capture larger eigenvalues.

### 6.5.2 3-D Wave Propagation




(a) (b)

Fig. 6.7. Wave propagation in a 3-D free space: (a) Waveform of electric fields at two observation points. (b) *Entire* solution error v.s. time.

The second example is a wave propagation problem in a 3-D free space. The same incident field is used as that of the first example. We also supply an exact absorbing boundary condition to all the unknowns on the boundary. Unlike that in the first example that has a abruptly changed grid size, a progressively changed grid size is adopted for space discretization. Along $y$- and $z$- direction, the space step is 0.1 m, and there are 5 cells. Along $x$- direction, there are 13 cells each having 0.1 m space step except for the three cells in the middle whose space step is 0.01 m, 0.001 m and 0.01 m respectively.

The existence of fine cells renders the time step of a conventional FDTD less than $1.07 \times 10^{-11}$ s. In contrast, the proposed method is able to use a time step solely determined by accuracy, which is $2.0 \times 10^{-10}$ s. As shown in Fig. 6.7(a), the electric fields obtained from the proposed method at two points located at (0.51, 0.45, 0.2) m and (0.57, 0.4, 0.2) m agree very well with analytical solutions. In Fig. 6.7(b), we assess the entire solution error measured by $\|\{e\} - \{e\}_{anal}\|/\|\{e\}_{anal}\|$, where $\{e\}$ consists of all 1,308 **E** unknowns obtained from the proposed method, while $\{e\}_{anal}$

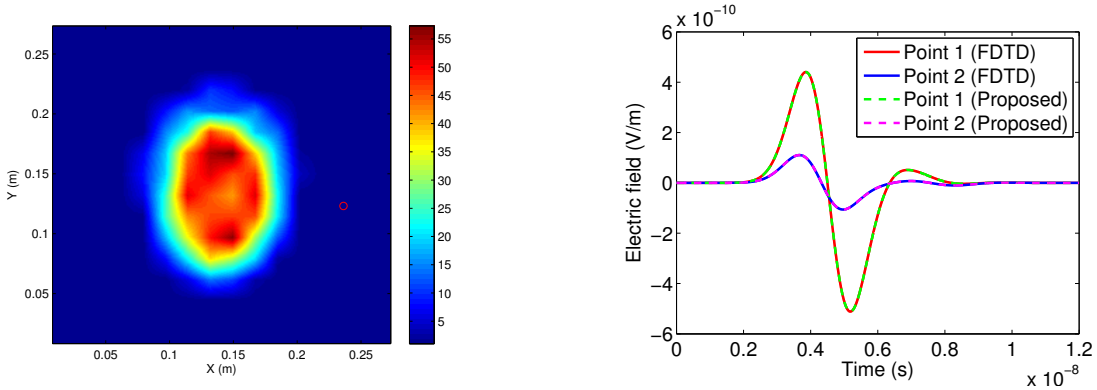
is analytical result. As can be seen clearly, the proposed method is accurate at all points, and across the whole time window simulated. The larger errors at early and late time are because the denominator of the solution error is zero at those times.

In this simulation, 125 cells are identified as fine cells, and the number of internal patches involved is 350. The size of $\mathbf{A}$ ($\mathbf{B}$) shown in (6.25) is 320 by 350. Given $\epsilon = 10^{-2}$, we obtain 120 unstable eigenmodes accurately from $\mathbf{S}_f$. It takes the proposed method 0.6470 s to finish the simulation. To simulate the same example, conventional FDTD costs 2.1608 s. The state-of-the-art unconditionally stable explicit FDTD method in [18] takes 0.3629 s to find the unstable modes, and 1.2545 s for explicit time marching. Hence, the propose method is faster than not only conventional FDTD, but also the method of [18]. This is because the method of [18] needs to deal with a global $\mathbf{S}$ matrix of size 1308 by 1308 to find the largest 120 unstable modes. In addition, the resultant $\mathbf{V}_h$ is dense, whereas the $\mathbf{V}_h$ in this method is zero in coarse cells, thus speeding up the explicit time marching step as well.

### 6.5.3 Inhomogeneous 3-D Phantom Head Beside a Wire Antenna

Previous examples are in free space, the third example is a large-scale phantom head [57] beside a wire antenna, which involves many inhomogeneous materials. The permittivity distribution of the head at $z = 2.8$ cm is shown in Fig.6.8(a). The wire antenna is located at (24.64, 12.32, 13.44) cm, the current on which has a pulse waveform of $\mathbf{J} = 2(t-t_0)e^{-(t-t_0)^2/\tau^2}$ with $\tau = 1.0\times10^{-9}$ s and $t_0 = 4\tau$. The size of the phantom head is 28.16 cm $\times$ 28.16 cm $\times$ 17.92 cm. The coarse step size along $x$-, $y$-, $z$-direction is 17.6 mm, 17.6 mm and 1.4 mm respectively, which results in $109,667$ unknowns. To capture the fine tissues located at the center of this head, three layers of fine grid whose length is 1.4 $\mu$m are added in the middle along $z-$direction. As a result, the conventional FDTD method can only use a time step less than $5.39\times10^{-15}$ s to ensure stability. In the proposed method, 768 fine cells are identified, which involve 4,709 electric field unknowns and 4,256 magnetic field unknowns. Given $\epsilon = 10^{-7}$,

1,088 unstable eigenmodes are obtained accurately from $\mathbf{S}_f$. With the contribution of unstable eigenmodes removed, the time step is increased to $2.56 \times 10^{-13}$ s. In Fig. 6.8(b), the electric fields at two points (12.32, 3.52, 13.44) cm and (12.32, 24.64, 13.44) cm are plotted in comparison with reference FDTD results. Again, very good agreement is observed. As for CPU time, the proposed method takes 84.8142 s to extract unstable eigenmodes, and 2895.7305 s for explicit time marching. However, the conventional FDTD needs 29968.7009 s to finish the same simulation. Meanwhile, although the method developed in [18] can also boost the time step up to the same value as the proposed method, it requires 8268.2 s instead in CPU time. Therefore, the proposed method is not only much faster than the conventional FDTD method, it is also more efficient than [18] since the proposed method requires the fine region only instead of the entire computational domain to extract unstable eigenmodes.




(a) (b)

Fig. 6.8. Simulation of a phantom head beside a wire antenna: (a) Relative permittivity distribution in a cross section of the phantom head at $z = 2.8$ cm. (b) Simulated electric field at two observation points in comparison with reference FDTD solutions.

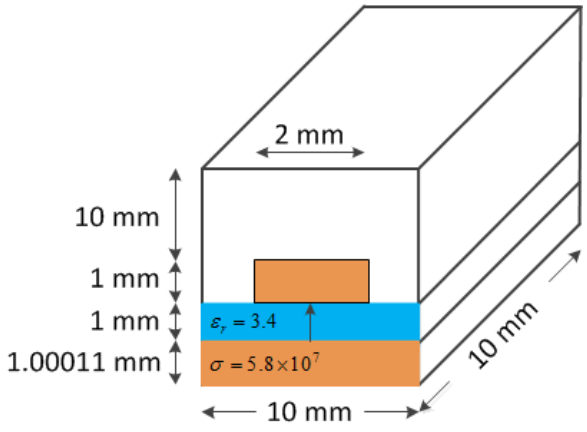
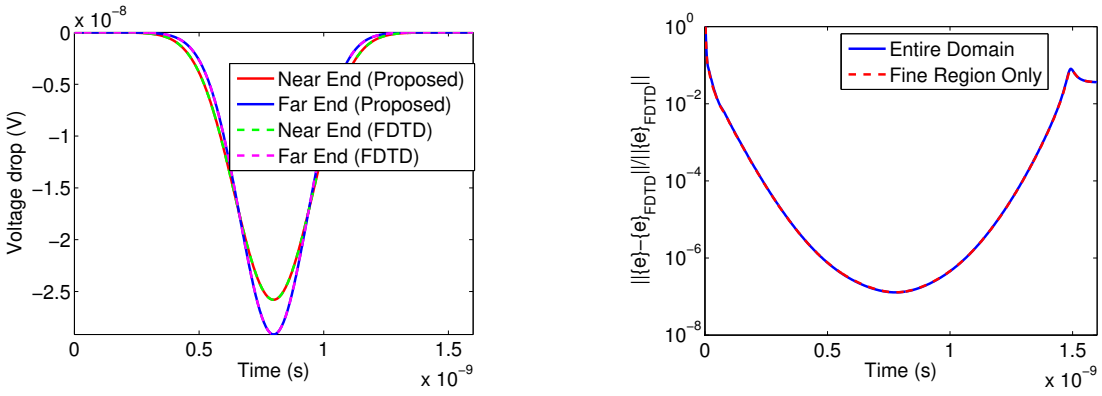



Fig. 6.9. Simulation of a microstrip line excited by a current source: Microstrip line structure.




(a) (b)

Fig. 6.10. Simulation of a microstrip line excited by a current source: (a) Simulated voltages at two ports. (b) Solution error in comparison with reference FDTD solutions in both entire domain and fine region only.

### 6.5.4 Inhomogeneous and Lossy 3-D Microstrip Line Structure

The last example is a microstrip line with lossy conductors and inhomogeneous dielectrics illustrated in Fig. 6.9. The details of the front view in x-y plane can be seen in Fig. 6.9 and the structure is 10 mm long in z-direction. A current source $\mathbf{J} = \hat{x}2(t-t_0)e^{-(t-t_0)^2/\tau^2}\mathrm{A/m}^2$ is launched between the bottom plate and the strip,

with $\tau = 2.0 \times 10^{-10}$ s and $t_0 = 4\tau$. The space step is 1 mm in all directions, but in order to capture skin effects, the second and the third space step in $x$ direction are chosen to be 0.1 $\mu$m, and 0.01 $\mu$m respectively. The total number of **E** unknowns in this structure is 5,335. Due to the small step size to capture skin effects, a time step of $1.35 \times 10^{-16}$ s is required in the conventional FDTD method. In contrast, the proposed method is able to use a time step of $8.7 \times 10^{-13}$ s. The number of terms kept in (6.45) is 9. In Fig. 6.10(a), the voltage drops extracted at both near and far ends of the strip line are plotted in comparison with the results obtained from a conventional FDTD method. It is clear to see that the simulated results agree very well with the reference solutions. To evaluate the entire solution error of the proposed method, we use a backward difference scheme in the conventional FDTD method with the same time step used in the proposed method, and store the solution at every time instant. The entire solution error at each time instant is calculated as $\|\{e\} - \{e\}_{FDTD}\| / \|\{e\}_{FDTD}\|$, and is plotted in Fig. 6.10(b). Obviously, the proposed method is accurate not only at the two sampled points, but also in the entire computational domain across the entire time window. Meanwhile, to evaluate the solution accuracy in fine region only, we also calculate the solution error for the unknowns residing in the fine region only. The error is also shown in Fig. 6.10(b), and it is almost the same as that in the entire computational domain.

In this simulation, not only the fine cells but also the cells filled with conductive materials are considered to form $\mathbf{M}_f$. Those cells involve 1,449 **E** unknowns and 1,352 **H** unknowns. The proposed method takes 49.2713 s to extract 1380 unstable modes for the prescribed accuracy $\epsilon = 10^{-6}$, and 437.7509 s for explicit marching, thus a total time of 487.0222 s. In contrast, the conventional FDTD based on (6.13) needs 5397.8587 s to finish the same simulation.

## 6.6 Conclusion

In this chapter, a fast explicit and unconditionally stable FDTD method requiring no global eigenvalue solution is developed. In this method, first we derive a new patch-based single-grid FDTD formulation, which naturally decomposes the curl-curl operator into a series of rank-1 matrices. This formulation helps us identify the relationship between fine cells and unstable eigenmodes. We find that the largest eigenmodes of the system matrix obtained from the entire computational domain can be accurately extracted from the system matrix assembled from the fine cells. The larger the contrast ratio between the fine-cell size and the coarse one, the more accurate the extracted eigenmodes. As a result, once there is a difference between the time step required by accuracy and that dictated by stability, the unstable modes can be extracted from fine cells. Based on this theoretical finding, we develop an accurate and fast algorithm for finding unstable modes from fine cells. We then upfront eradicate these unstable modes from the numerical system before performing an explicit time marching. The resultant simulation retains the merit of the original explicit FDTD in avoiding solving a matrix equation, while eliminating its shortcoming in time-step's dependence on space step. The proposed method is also extended to handle general lossy problems where dielectrics and conductors are inhomogeneous and lossy. Numerical experiments including both lossless and lossy problems have demonstrated the accuracy, efficiency, and unconditional stability of the proposed method, by comparing with conventional FDTD as well as the state-of-the-art explicit and unconditionally stable methods.

It is also worth mentioning that although the unstable modes are extracted from fine cells and subsequently removed for a stable simulation, this does not mean that the resultant field solution in the fine cells is zero or has a large error. This is because the stable eigenmodes preserved in the numerical system have their field distributions all over the grid, including both coarse and fine cells. The unstable eigenmodes are discarded because their contributions to the field solution is negligible in coarse as

well as fine regions. Notice that the weight of an eigenmode in the field solution is inversely proportional to the distance between its eigenvalue and the square of the working frequency, irrespective of fine or coarse regions.

When the contrast ratio between fine cells and regular cells is small such as less than 2, the speedup of the proposed method may be little because of additional computational overhead for finding the unstable modes. The accuracy is also lower for smaller contrast ratio as compared to larger contrast ratio in space step. But good accuracy can still be obtained for small contrast ratio. The error is also well controlled by checking $\epsilon_{acc}$ in (6.20). If the fine-cell region results in a matrix of large size, it may become expensive to extract all of the unstable modes from the fine-cell region, although the method is still more efficient than that in [18] where the entire grid thereby system matrix is handled for obtaining the unstable modes. In this case, one can obtain a subset of the largest eigenmodes from the fine cells rather than all of them to enlarge the time step to a certain extent, instead of all the way up to that permitted by accuracy. In addition, the combination of the proposed method with the efficient method for finding stable modes such as that in [16, 17] may also be a better option in some applications.

# 7. AN UNSYMMETRIC FDTD SUBGRIDDING ALGORITHM WITH UNCONDITIONAL STABILITY

## 7.1 Introduction

The finite-difference time-domain method is one of the most popular time-domain methods for electromagnetic analysis [1]. This is mainly because of its simplicity and optimal computational complexity at each time step. The conventional FDTD method requires a uniform orthogonal grid. If there exist fine features in a structure, a fine space step must be used to discretize them. Because of the connected nature of an orthogonal grid, the regions where there are no fine features are also discretized in a smaller space step. This unnecessarily increases the number of unknowns to be solved. Subgridding is an effective means to address this problem, where fine grids are only placed in the necessary regions, which do not need to be conformal to the background regular grid.

In an FDTD subgridding method, the fields at the interface between coarse and fine meshes are typically estimated through certain interpolation scheme. Such an interpolation may ruin the positive semi-definiteness of the original FDTD numerical system, thereby causing instability. Meanwhile, the numerical reflections at the interface between coarse and fine meshes and the different numerical dispersion in the two meshes may result in a worse solution accuracy. Therefore, a good FDTD subgridding algorithm should guarantee both stability and accuracy.

In literature, extensive work has been done to tackle the FDTD subgridding problem. In [58], an initial run is made on a coarse grid, the result of which is then used as the boundary condition for a second calculation where the grid in the region of interest is refined. Later, a variable step size method (VSSM) was developed in [19]. It provides a direct interpolation scheme to update fields in both coarse and fine

grids simultaneously when a grid contrast ratio is 2. It also develops an interpolation scheme based on the wave equation for a contrast ratio of 3. The wave equation based scheme was improved to be a mesh refinement algorithm (MRA) in [20] by interpolating a second-order difference at each mesh node, and later extended to be a multigrid displacement method (MGDM) by adding a buffer zone between coarse and fine meshes in [21]. To handle material traverse, a new subgridding algorithm in [59] was developed for odd contrast ratios. Later, a multigrid current method (MGCM) was proposed in [22] to handle any contrast ratio by using a weighted current value from the coarse region at the mesh interface to update the fine-region tangential fields on the same interface. To minimize the numerical reflection, in [60], the authors proposed a new arrangement of mesh where the coarse and fine mesh are offset in all directions. Such a mesh allows the development of a pulsing overlapping scheme where the outermost layer of the fine mesh is dropped during update, but the mesh is expanded back to its original size at the end of each update cycle. Instability especially late-time instability has been observed in many of the aforementioned subgridding algorithms. Various approaches have been proposed to remedy this issue [21,22,59,60]. However, they still lack a theoretical study on the stability. In [61], a subgridding scheme with reciprocal interpolation scheme was developed in a recessed subgridding interface with stability guaranteed, but the solution accuracy is compromised.

In [62, 63], a class of subgridding algorithms was developed in the framework of the finite integration technique (FIT) and the stability of this method is controlled by maintaining the consistency of the field coupling scheme. It handled cases where the contrast ratio is 2. Another subgridding method based on the finite element method (FEM) was proposed in [64]. The concept of maintaining the consistency of the field coupling scheme can also be found in [65], which is based on an equivalent passive network method. All of these methods involve a hybridization with other methods.

Among existing FDTD subgridding algorithms, the consistency of the field coupling scheme or reciprocity has been widely adopted as a viable means to ensure stability. In other words, if a field unknown A is used to generate a field unknown

B, then the field unknown B should also be involved in the generation of the field unknown A. In some algorithms, the coupling coefficient from A to B, and vice versa are also enforced to be equal. This certainly limits the accuracy of the interpolation schemes as well as the meshing flexibility in a subgridding scheme.

In this work, to systematically control the stability of the FDTD subgridding algorithm without sacrificing accuracy, we first reformulate the FDTD algorithm from the original edge-based dual-grid one to a new patch-based single-grid formulation. Using this formulation, we only need to generate one column vector and one row vector for each patch in a single grid, regardless of whether the grid is 2-D or 3-D, it has subgrids or not, and the grid/subgrid is uniform or non-uniform. The product of the column vector and the row vector of each patch is a rank-1 matrix. The system matrix is simply the sum of the rank-1 matrices. Based on this new representation of the FDTD algorithm, the stability of the FDTD-based methods can be readily analyzed for both regular grids and grids having subgrids. In a regular grid, each rank-1 matrix comprising the FDTD system matrix is positive semi-definite, and hence the sum of them remains to be positive semi-definite, thus ensuring the stability. In other words, one can always find a time step to make the explicit FDTD time marching stable. However, when subgrids are present, since field unknowns at the interface would have to be interpolated from adjacent unknowns to ensure accuracy, the resultant rank-1 matrix is usually unsymmetrical. When the unsymmetrical matrix has complex-valued or negative eigenvalues, it will make a traditional explicit marching absolutely unstable. However, in general, we cannot rule out these eigenvalues from an unsymmetrical matrix. Even though the unsymmetrical matrix generated from each patch has non-negative real eigenvalues, we cannot prove the sum of these unsymmetrical matrices has non-negative real eigenvalues only. The property of a symmetric matrix does not apply to an unsymmetrical matrix. To overcome this problem, we propose a new time marching scheme, which preserves the FDTD's advantage in matrix-free time marching, while remaining to be stable in the presence of complex and negative eigenvalues. As a result, the proposed method does not require

reciprocal operations from one field unknown to the other to guarantee stability. The proposed time marching scheme is also general, which can be used to make other unsymmetrical FDTD subgridding algorithms stable.

With the stability guaranteed in time, the interpolation schemes can be developed solely to ensure accuracy. We hence develop an accurate interpolation scheme to ensure the accuracy of the resulting subgridding algorithm. This scheme is applicable to arbitrary contrast ratios between the normal gird and the subgrid, as well as supporting non-uniform subgridding. We also show that since there are only a few kinds of rank-1 matrices in the proposed algorithm, the maximum time step permitted for a stable simulation can be analytically analyzed. The proposed subgridding algorithm is then further made unconditionally stable, based on our prior work in [66]. Extensive numerical experiments involving both 2- and 3-D subgrids with various contrast ratios have demonstrated the accuracy, stability, and efficiency of the proposed new subgridding method.

## 7.2 Comparison between FDTD without Subgrids and with Subgrids

In the original FDTD algorithm, one field unknown is placed in a primary grid at the center point of each edge, and also tangential to the edge. The other field unknown is placed in a dual grid in the same way. If there are $N_e$ electric field unknowns, and $N_h$ magnetic field unknowns, then there are $N_e + N_h$ equations in the FDTD-based discretization of Maxwell's equations. Essentially, we can view each equation is written for obtaining one electric or magnetic field unknown. For example, obtaining the time derivative of one electric field unknown from its surrounding magnetic field unknowns, and vice versa.

When there is a subgrid present in the discretization, the original FDTD algorithm has to be modified. There are also subgridding techniques that are not purely based on FDTD anymore. However, if still using the original framework of FDTD, on the interface between the normal grid and the subgrid, one would face the following

problem. The generation of the primary field unknown would require the dual field unknown at the points that are not coincident with the points where the dual field is generated from the primary field. A natural remedy to this problem is to interpolate the unknown dual field at the desired point from the known dual fields at adjacent points. Such an interpolation scheme is not unique. However, its effect on accuracy and stability is different. A theoretical stability analysis is still lacking in many subgridding algorithms. On the other hand, late-time instability has been observed from many existing techniques. When instability occurs, there is no fundamental way forward to correct the stability problem.

Next, we will first present the proposed theory for making an FDTD subgridding algorithm stable in general subgrid settings. We then proceed to the details of the proposed subgridding method.

## 7.3 Proposed Theory

### 7.3.1 Reformulating FDTD Based on Patches in a Single Grid

To facilitate the development of a subgridding algorithm, we propose to first reformulate the FDTD into a different format. If we term the original FDTD formulation an edge-based dual-grid formulation (as each edge in the primary and dual grid is associated with one field unknown), this alternative formulation is a patch-based single-grid formulation. In the original formulation, since an edge-based approach is used together with dual grids, when there are subgrids, there are many scenarios to consider. In contrast, the proposed new formulation is based on patches in a single grid. As a result, the subgridding scenarios to be considered become only a few kinds.

We use only one grid. In this grid no matter it is a 2-D or 3-D grid, we loop over all the patches present in the grid. For each patch, we formulate a column vector and a row vector, whose product is a rank-1 matrix. The row vector describes how the $\mathbf{E}(\mathbf{H})$ unknowns along the contour of the patch produce the normal $\mathbf{H}(\mathbf{E})$ field at the patch center. The column vector describes how the normal $\mathbf{H}(\mathbf{E})$ field at the

patch center is used to obtain the $\mathbf{E}(\mathbf{H})$ unknowns. The two are transpose of each other in a uniform grid, but can be very different in a non-uniform grid or a grid with subgrids. For example, with subgridding, the normal $\mathbf{H}(\mathbf{E})$ field at the patch center may have to be used to obtain the $\mathbf{E}(\mathbf{H})$ unknowns elsewhere not belonging to the same patch. With the two vectors generated for each patch, we can march on in time to find the electric and magnetic field solutions. We can also add the rank-1 matrix of each patch, and obtain a second-order differential equation in time to perform time marching. In the following presentation of the proposed formulation, we place the normal $\mathbf{H}$ at the patch center, and $\mathbf{E}$ along the edges of the grid. But the two can also be reversed.

Consider a general 2-D or 3-D grid. For each patch, based on the FDTD algorithm, we obtain the magnetic field normal to the patch at the patch center, $h_i$, as the following:

$$\left[-\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}\right] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = -\mu_i \frac{\partial h_i}{\partial t}, \tag{7.1}$$

where subscript $i$ denotes the patch index, $e$ denotes the tangential electric field at the center point of every edge in the patch, as illustrated in Fig. 7.1. The $L_i$ and $W_i$ are, respectively, the two side lengths of patch $i$, and $\mu_i$ is the permeability at the patch center. (7.1) can be rewritten as

$$[b]_i^T [e]_i = -\mu_i \frac{\partial h_i}{\partial t}, \tag{7.2}$$

where $[e]_i$ denotes the column vector containing all of the electric field unknowns of patch $i$, and $[b]_i^T$ is a row vector of

$$[b]_i^T = \left[-\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}\right]. \tag{7.3}$$

Let $\{e\}$ be a vector consisting of all $N_e$ electric field unknowns in a grid, (7.2) can be rewritten as

$$\{b\}_i^T \{e\} = -\mu_i \frac{\partial h_i}{\partial t}, \tag{7.4}$$

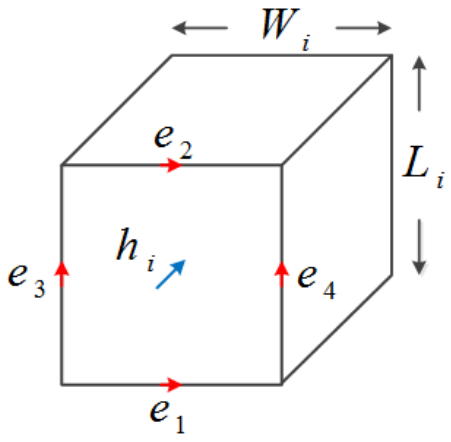



Fig. 7.1. Illustration of a patch-based discretization of Faraday's law.

in which $\{b\}_i$ is $[b]_i$ in (7.3) extended to length $N_e$ such that $\{b\}_i^T\{e\} = [b]_i^T[e]_i$. Obviously, $\{b\}_i$ has only four nonzero entries as follows

$$\{b\}_i(g(i,k)) = [b]_i(k), \quad k = 1,2,3,4. \tag{7.5}$$

in which $g(i,k)$ denotes the index of the $k$-th electric field unknown of patch $i$ in the global electric field vector $\{e\}$. Consider all patches present in the mesh, the discretization of Faraday's law can be represented as

$$\mathbf{S}_e\{e\} = -diag\{\mu\}\frac{\partial\{h\}}{\partial t}, \tag{7.6}$$

where $\{h\}$ contains all of the $h$ unknowns whose number is $N_h$, $diag\{\mu\}$ is a diagonal matrix of permeability. $\{b\}_i^T$ is the $i$-th row of $\mathbf{S}_e$.

In a general patch present in a grid with subgridding, the row vector shown in (7.3) will be different. But its entries remain to be the weighting coefficients of the electric field unknowns along the contour of a patch for generating the normal magnetic field at the patch center. To be more specific, $[b]_i$ has $m$ entries, where $m$ is the number of electric field unknowns along the contour of patch $i$. An arbitrary $k$-th entry of $[b]_i$, $[b]_i(k)$, is simply the weighting coefficient of electric field unknown $e_k$ used to generate $h_i$. Its sign is determined by the right hand rule. With the right-hand thumb pointing

to the direction associated with $h_i$, if $e_k$'s direction is along the direction encircling the $h_i$'s direction, the sign is positive. Otherwise, the sign is negative.

In the original FDTD formulation, the discretization of Ampere's law is performed on a dual grid, resulting in the following matrix equation

$$(\mathbf{S}_h)_{N_e \times N_h} \{h\} = diag\{\epsilon\} \frac{\partial \{e\}}{\partial t} + \{j\}, \tag{7.7}$$

where $\{h\}$ contains all of the $h$ unknowns whose number is $N_h$, $diag\{\epsilon\}$ is a diagonal matrix of permittivity, and $\{j\}$ denotes a current source vector. Each row of the above equation simply denotes a discretized curl operation performed on the magnetic fields producing the time derivative of an electric field.

In the proposed alternative formulation, we rewrite (7.7) as the following:

$$\{a\}_1 h_1 + \{a\}_2 h_2 + ... + \{a\}_{N_h} h_{N_h} = diag\{\epsilon\} \frac{\partial \{e\}}{\partial t} + \{j\}, \tag{7.8}$$

where the matrix-vector multiplication of $\mathbf{S}_h\{h\}$ in (7.7) is realized as the sum of weighted columns, instead of the traditional row-based computation which we are more familiar with. Here, the $\{a\}_i$ is simply the $i$-th column of $\mathbf{S}_h$, and $h_i$ is the $i$-th entry of vector $\{h\}$, which is nothing but the normal magnetic field at the center of patch $i$. Based on how Ampere's law is discretized in the FDTD method, it is evident that $\{a\}_i$ has only nonzero entries at the rows whose indexes correspond to the electric field unknowns generated from $h_i$. In a regular grid, $h_i$ is used to generate four electric field unknowns, which are those along the four sides of patch $i$. Hence, $\{a\}_i$ has only four nonzero elements, with all the others being zero. Removing the zeros, $\{a\}_i$ simply becomes a vector of length four in each patch as the following:

$$[a]_i = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i} \\ \frac{1}{W_i} \\ -\frac{1}{W_i} \end{bmatrix}. \tag{7.9}$$

Clearly, it is the same as $[b]_i$ in a uniform grid.

In a general patch present in a grid with subgridding, the column vector $[a]_i$ can become different from that shown in (7.9). However, its entries remain to be the weighting coefficients of the magnetic field used to generate the electric field unknowns. To be more specific, an arbitrary $k$-th entry of $[a]_i$, $[a]_i(k)$, is simply the weighting coefficient of $h_i$ used to generate $e_k$.

Though mathematically identical to (7.7), (7.8) allows us to discretize Ampere's law in the original grid of $\mathbf{E}$ and use the same patch-based approach. Basically, to discretize the Ampere's law, we also loop over all the patches in the original grid. On each patch, we generate a column vector $\{a\}_i$ ($i = 1, 2, ..., N_h$). Scaling $\{a\}_i$ by $h_i$ and summing it up over all the patches in the original grid, we obtain the discretization of the curl of $\mathbf{H}$, as shown by the left-hand side of (7.8).

Now, if we take a time derivative of (7.8), and substitute (7.4) into it, we obtain

$$\sum_{i=1}^{N_h} \left( \frac{1}{\mu_i} \{a\}_i \{b\}_i^T \right) \{e\} = -diag\{\epsilon\} \frac{\partial^2 \{e\}}{\partial t^2} - \frac{\partial \{j\}}{\partial t}, \tag{7.10}$$

which can be compactly written as

$$\frac{\partial^2 \{e\}}{\partial t^2} + \mathbf{C}\{e\} = -diag\{\frac{1}{\epsilon}\} \frac{\partial \{j\}}{\partial t} \tag{7.11}$$

where

$$\mathbf{C} = diag\{\frac{1}{\epsilon}\} \sum_{i=1}^{N_h} \frac{1}{\mu_i} \{a\}_i \{b\}_i^T, \tag{7.12}$$

which is clearly the sum of the rank-1 matrix obtained from each patch.

In the proposed patch-based formulation, after $[a]_i$ and $[b]_i$ are obtained for each patch, we can use them to perform a leap-frog time marching based on (7.4) and (7.8). We can also directly solve (7.11) as a second-order differential equation in time. Since a single grid is used, and the two vectors can be generated for each patch individually, the new formulation makes it much easier to develop FDTD subgridding algorithms. It actually also makes the original FDTD simpler for implementation in a uniform grid.

### 7.3.2 Stability Analysis of FDTD without and with Subgrids

The stability of the first-order systems (7.4) and (7.8) as well as the second-order based (7.11) is determined by the following eigenvalue problem

$$\mathbf{C}x = \lambda x. \tag{7.13}$$

To analyze the stability, we can expand the field solution $\{e\}$ by using the eigenvectors of (7.13), obtaining

$$\{e\} = \mathbf{V}\{y\}, \tag{7.14}$$

where $\mathbf{V}$ denotes a matrix whose columns are eigenvectors. Substituting (7.13) into (7.11), and multiplying both sides of (7.11) by $\mathbf{V}^T$, we obtain

$$\mathbf{V}^T\mathbf{V}\frac{\partial^2\{y\}}{\partial t^2} + \mathbf{V}^T\mathbf{C}\mathbf{V}\{y\} = 0, \tag{7.15}$$

where source is removed as it is irrelevant to the stability analysis. Since $\mathbf{V}^T\mathbf{C}\mathbf{V} = \mathbf{V}^T\mathbf{V}\Lambda$, where $\Lambda$ is the diagonal matrix of eigenvalues $\lambda_i$, (7.15) becomes

$$\frac{\partial^2 y_i}{\partial t^2} + \lambda_i y_i = 0, \quad (i = 1, 2, ..., N_e) \tag{7.16}$$

Performing a $z$-transform of the above, if all the eigenvalues $\lambda_i$ are non-negative real, a time marching based on central difference scheme would be stable as long as

$$\Delta t < \frac{2}{\sqrt{\lambda_{max}}}, \tag{7.17}$$

where $\lambda_{max}$ is the largest eigenvalue. In this case, (7.11) can be marched on in time explicitly as

$$\{e\}^{n+1} = \left(2 - \Delta t^2\mathbf{C}\right)\{e\}^n - \{e\}^{n-1} - \Delta t^2 diag\{\frac{1}{\epsilon}\}\left(\frac{\partial\{j\}}{\partial t}\right)^n \tag{7.18}$$

However, when the eigenvalues of $\mathbf{C}$ are complex-valued or negative, no time step can make (7.16) stable [50]. In an FDTD subgridding scheme, since interpolations are used to obtain the unknown fields at the subgrid interfaces, the resulting rank-1 matrix of each patch is not symmetric. The same is true for the global system

matrix assmebled from each patch's contribution. An unsymmetric matrix can have complex-valued eigenvalues or even negative ones. In many cases, one can prove the eigenvalues of a Maxwell's system to be non-negative if they are real. However, the complex eigenvalues cannot be ruled out, in general. This can also be numerically verified. When this happens, an FDTD subgridding algorithm is absolutely unstable.

### 7.3.3 How to Guarantee Stability When the System Matrix is Unsymmetric?

The aforementioned stability problem for an unsymmetric matrix can be resolved by first employing a backward difference scheme to discretize (7.11) as follows

$$\left(\mathbf{I} + \Delta t^2 \mathbf{C}\right) \{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} \tag{7.19}$$

$$- \Delta t^2 diag\{\frac{1}{\epsilon}\} \left(\frac{\partial \{j\}}{\partial t}\right)^{n+1}. \tag{7.20}$$

Since a backward difference scheme is unconditonally stable, we are allowed to use an arbitrarily large time step. However, by doing so, we have to solve a system matrix of $(\mathbf{I} + \Delta t^2 \mathbf{C})$. To retain the matrix-free merit of the FDTD, we can choose the following time step to perform the backward time marching

$$\Delta t < \frac{1}{\sqrt{\lambda_{max}}}. \tag{7.21}$$

With the above, $\|\Delta t^2 \mathbf{C}\| = |\Delta t^2 \lambda_{max}| < 1$ is satisfied. Hence, the inverse of $\mathbf{I} + \Delta t^2 \mathbf{C}$ becomes explicit, which can be evaluated as

$$\left(\mathbf{I} + \Delta t^2 \mathbf{C}\right)^{-1} = \mathbf{I} - \Delta t^2 \mathbf{C} + \left(\Delta t^2 \mathbf{C}\right)^2 - \ldots. \tag{7.22}$$

The above series can be truncated at the $k$-th term without sacrificing accuracy, where $k$ is usually less than 10 as (7.21) is satisfied. Since (7.22) does not involve any matrix inversion, we can still obtain the solution in (7.19) explicitly as the following:

$$\{e\}^{n+1} = \left(\mathbf{I} - \Delta t^2 \mathbf{C} + \ldots + \left(\Delta t^2 \mathbf{C}\right)^k\right) \{f\}, \tag{7.23}$$

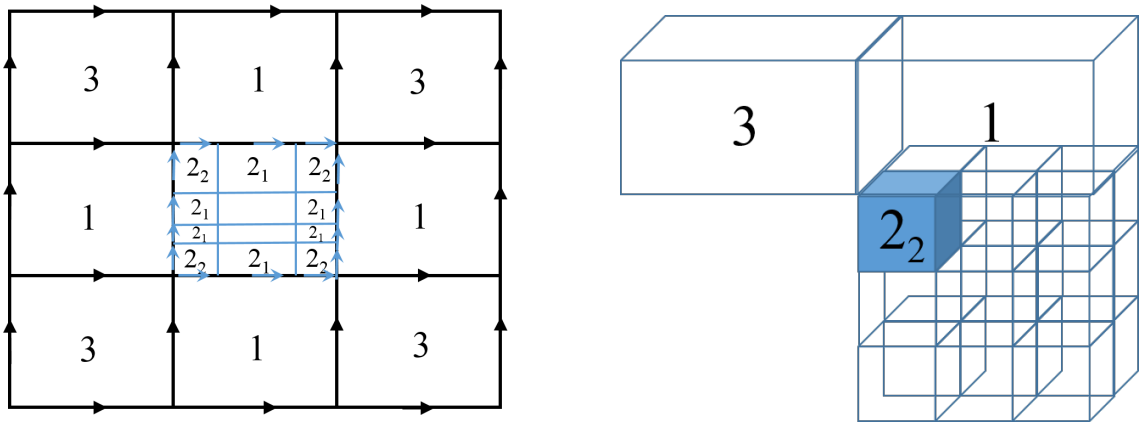
where $\{f\}$ denotes the terms moved to the right hand side.

Therefore, no matter whether the system matrix $\mathbf{C}$ is symmetric or not, we can find the solution explicitly via either (7.18) or (7.23) without incurring any instability. More importantly, the choice of the time step shown in (7.21) also agrees with the choice of the time step of a traditional explicit time marching. Hence, we do not sacrifice in the size of time step, while making the inverse of the backward-difference based system matrix explicit.

## 7.4 Proposed Subgridding Algorithm with Guaranteed Stability and Accuracy

In an FDTD grid with subgrids, the patches can be categorized into two big classes. One has its regular $[a]$ and $[b]$ vectors. The other class of patches have modified $[a]$ and $[b]$ vectors, because the fields along the subgrid edges have to be obtained through interpolations across patches to ensure accuracy. Based on the stability analysis in Section 7.3.2, it is not necessary to have the two curl operators to be reciprocal to guarantee stability, thus the interpolation scheme can be made very flexible. Since the field solution in the FDTD algorithm is known along three orthogonal directions in an orthogonal grid, the interpolation can be carried out in three directions to achieve good accuracy. In this section, we develop a novel FDTD subgridding algorithm with guaranteed accuracy. This algorithm supports an arbitrary contrast ratio of the regular grid size to the subgrid size. It also allows for non-uniform grids in both regular and subgrid regions.

Consider a regular grid involving subgrids as shown in Fig. 7.2(a) and 7.2(b), we place all of the electric field unknowns along the edge of the grid and at the center of each edge. Thus, our $\{e\}$ is composed of tangential electric field along each edge in the regular grid (regular edge), in the subgrid (subgrid internal edge), and on the interface between the regular grid and the subgrid (subgrid interface edge), as illustrated in Fig. 7.2(a). The magnetic field unknown is placed at the center of each patch, along the normal direction of the patch. Thus, $\{h\}$ consists of the




(a) (b)

Fig. 7.2. Illustration of a grid with subgrids. (a) 2-D. (b) 3-D.

magnetic fields normal to each patch at the patch center. It is also worth mentioning that although both positive and negative directions can be chosen as the reference direction of the electric field unknown along each edge, we choose the conventional positive $x$-, $y$-, and $z$-directions. The same is true for the reference direction of the normal magnetic field at the patch center.

### 7.4.1 Building Column Vector $[a]$ and Row Vector $[b]^T$ for Each Patch with Guaranteed Accuracy

A grid can involve many patches. However, we find that regardless of a 2-D or 3-D grid, the patches can be categorized into three irregular types based on their corresponding $[a]$ and $[b]$ vectors. This is attributed to the proposed patch-based formulation, which makes the resultant subgridding algorithm suitable for both 2-D and 3-D grids with almost no change. Next, we elaborate the construction of $[a]$ and $[b]$ vectors for each type of the irregular patches.

**Irregular patch type 1**

This patch is a coarse patch in the regular-grid region, but having at least one side shared with the subgrid region, as shown by the patches marked as 1 in Fig. 7.2(a) and 7.2(b). For convenience of explanation, we consider one side with subgridding. Along this side, there are more than one edges due to subgridding. Let the number of edges on this side be $n$, and the length of the $j$-th edge be $l_j$. The $l_j$ can be the same for all edges. It can also be different in different edges, as illustrated in Fig. 7.2(a).

To generate the magnetic field at the coarse patch center, we need to use the tangential electric field at the center of each side. For the side having subgrids, the electric field unknowns are placed at the center of each subgrid edge. Hence, the electric field at the center of the side needs to be obtained from the subgrid electric fields. This can be accurately done as the following

$$e_c = \sum_{j=1}^{n} \frac{l_j}{L_i} e_j, \tag{7.24}$$

in which $L_i$ is the entire length of the side, where subscript $i$ denotes the patch index. The resulting row vector $[b]_i^T$ for this patch can be written as

$$[b]_i^T = \left[ -\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i} v^T \right], \tag{7.25}$$

where

$$v^T = \left[ \frac{l_1}{L_i}, \frac{l_2}{L_i}, ..., \frac{l_n}{L_i} \right]. \tag{7.26}$$

Hence, the $[b]_i^T$ is no longer of length 4, but of length $3 + n$. The accuracy of the resulting (7.2) is of second order. This is because if we perform a line integral of the electric field along the contour of the patch using the electric field unknowns located on the contour, and equate it to $-\mu \frac{\partial h_i}{\partial t}$ at the patch center multiplied by the patch area, we will obtain (7.25).

The above $[b]_i^T$ is written for the case when the fourth electric field in a patch is associated with the subgrids. If it is another electric field, say the $j$-th electric field, the $v^T$ is multiplied to the $j$-th entry of the original $[b]_i^T$, and the denominator

of (7.26) should be changed to the length of the side having subgrids. If there are multiple sides shared with the subgrid region, then $v^T$ will be attached to each entry associated with the subgridding side.

To construct column vector $[a]_i$ for this patch, we need to find out how the magnetic field at this patch is used to generate electric field unknowns. Within this patch, the electric field unknown along the regular edge is obtained from the magnetic field at the center of this patch, and the other one at the center of the adjacent patch sharing the regular edge. Hence, the corresponding entry in $[a]_i$ is the same as that in a regular discretization, which is $\pm\frac{1}{L_i}$, $\pm\frac{1}{W_i}$, or another one if a non-uniform grid is used. However, to ensure accuracy, the electric field along the subgrid interface edge cannot be obtained in the same way. Take one subgrid interface edge highlighted by a red arrow in Fig. 7.3 as an example, to obtain the electric field accurately at the edge center, we need to know the magnetic field at the point marked by $\times$ above the red arrow. Since the magnetic fields are only known at the center of every patch, the magnetic field at this point has to be interpolated. Here, we perform a linear interpolation using the magnetic fields at adjacent patches since it can provide a second-order accuracy.

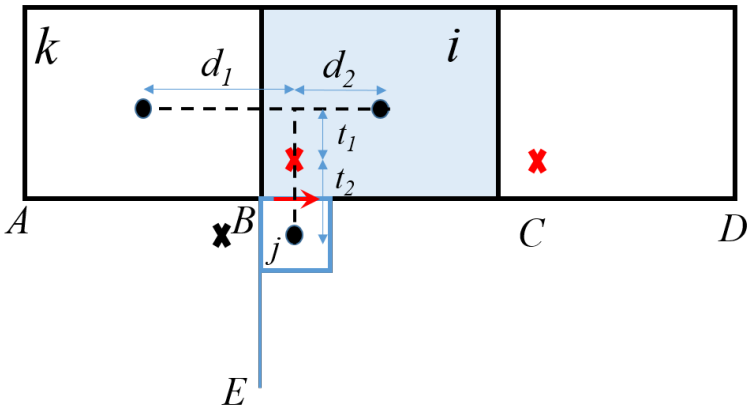



Fig. 7.3. Illustration of the interpolation scheme.

To explain this interpolation scheme, let the coarse patch being considered be patch $i$. The subgrid interface edge must be shared by patch $i$ and a fine patch in the subgrid. Let this fine patch be patch $j$, as illustrated in Fig. 7.3. Let the magnetic fields at the center points of the two patches be respectively $h_i^c$, and $h_j^f$, where we use the superscript to indicate whether the patch is coarse or fine. To interpolate the magnetic field at the marked point accurately, we also find another coarse patch $k$. This patch and patch $i$ shares a regular edge in common, and this regular edge is perpendicular to the subgrid edge, and closer to the subgrid edge in between the two regular edges of patch $i$. We denote the magnetic field at the center of this patch by $h_k^c$. The magnetic field at the marked point can then be accurately interpolated as

$$h_\times = \frac{t_2}{t}\left(\frac{d_2}{d}h_k^c + \frac{d_1}{d}h_i^c\right) + \frac{t_1}{t}h_j^f, \tag{7.27}$$

where $t_1$, $t_2$, $d_1$, and $d_2$ are distances labeled in Fig. 7.3, and $t = t_1 + t_2$, $d = d_1 + d_2$. These distances can be readily found from the coordinates of the three patch centers. Obviously, a linear interpolation along all directions is used to obtain the magnetic field at the marked point. With $h_\times$, the electric field at the subgrid interface edge can be accurately obtained from the magnetic field at the fine patch center, and that at the marked point as the following

$$\epsilon\frac{\partial e_j}{\partial t} = \frac{h_j^f - h_\times}{W_j^f}. \tag{7.28}$$

Substituting (7.27) into the above, obviously, the coefficient in front of $h_i$ for generating $e_j$ is $-\frac{1}{W_j^f}c_j$, where $c_j = \frac{t_2}{t}\frac{d_1}{d}$ and the distance parameters are those corresponding to the $j$-th subgrid edge.

The aforementioned interpolation results in the following $[a]_i$ vector

$$[a]_i = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i} \\ \frac{1}{W_i} \\ -u \end{bmatrix}, \tag{7.29}$$

in which $u$ is a vector of

$$u = \begin{bmatrix} \frac{c_1}{W_1^f} \\ \frac{c_2}{W_2^f} \\ \vdots \\ \frac{c_k}{W_k^f} \end{bmatrix}, \tag{7.30}$$

where $W_j^f$ is the width of the fine patch whose electric field is generated from $h_i$, and $c_j (j = 1, 2, ...k)$ are positive coefficients between 0 and 1. Here, $k$ can be greater than $n$ because the magnetic field at patch $i$ may also be used to obtain electric fields not belonging to patch $i$. To be specific, on the patch $i$ being considered, we can generate $n$ such $c$-coefficients, where $n$ is the number of subgrid edges on the side having subgrids. Take Fig. 7.3 as an example, this is the number of subgrid edges on side $BC$. The rest of $k - n$ entries in $u$ are due to other electric field unknowns generated from $h_i$. In the following, we will give a complete count of these electric field unknowns.

In a 2-D setting, if along the adjacent sides of $BC$, namely right half of $AB$ and left half of $CD$, there are subgrids, then the electric fields on these subgrid edges will have to be generated from $h_i$. This is because $h_i$ will be used to interpolate the missing magnetic field required to generate the electric fields on those edges, as highlighted by a red mark adjacent to $CD$. The same linear interpolation as shown in (7.27) can be used, from which the corresponding $c_j$ coefficient can be identified. In a 3-D setting, the three sides of $AB$, $BC$, and $CD$ become six patches perpendicular to patch $i$ and centering patch $i$, with three on one side of patch $i$; and the other three on the other side of patch $i$. All the subgrid edges on the six patches along the direction of $BC$ will be related to $h_i$. The electric field unknowns on these edges will be interpolated in the same way as illustrated in Fig. (7.27). If coarse patch $i$ and $k$ for the subgrid edge do not exist (this can happen for a subgrid edge falling onto the face of a subgrid region), the adjacent coarse patches parallel to the imaginary patch $i$ and $k$ can be used to interpolate magnetic fields at the center points of imaginary patch $i$ and $k$, and subsequently used in (7.27). The resultant $c_j$ coefficients in front of $h_i$ remain to

be between 0 and 1. Regardless of 2-D and 3-D, since the electric field unknowns to be generated from $h_i$ on patch $i$ are all orientated in the same direction, the sign of their corresponding entries in $[a]_i$ is the same. If there are multiple sides shared with the subgrid region in patch $i$, similarly, vector $u$ will appear at the corresponding entry, and follow the original sign of the entry.

**Irregular patch type 2**

For this type, the patch is a fine patch in the subgrid but with at least one side falling onto the subgrid interface with the regular grid. This type of patches is illustrated by patches marked by $2_2$ and $2_1$ in Fig. 7.2(a) and 7.2(b), where subscript denotes the number of edges on the interface.

In such a patch, the $[b]_i^T$ remains the same as that in a regular grid, but the length and width used are the fine-patch counterparts. Thus, we have

$$[b]_i^T = \left[-\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}\right]. \tag{7.31}$$

However, the $[a]_i$ is different. Again, to determine $[a]_i$, we need to find out how the magnetic field at this patch is used to generate electric field unknowns. Within the patch, among the four electric field unknowns, two are not located on the interface, and thereby shared by two fine patches. They are generated from the $h_i$ in the same way as the regular ones. For the two residing on the interface, each of them requires one magnetic field that is outside the subgrid and unknown, as shown by the marks in Fig. 7.3. Again, such a magnetic field is interpolated from the magnetic fields at the three patch centers in the same way as shown in (7.27). Hence, the resultant $[a]_i$ vector is

$$[a]_i = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i}(1-c_2) \\ \frac{1}{W_i}(1-c_3) \\ -\frac{1}{W_i} \end{bmatrix}, \tag{7.32}$$

where $c_2$ and $c_3$ are positive coefficients between 0 and 1. Based on (7.27), they have the form of $c_j = \frac{t_1}{t}$ in which $t_1$ and $t$ are distance parameters associated with the subgrid edge residing on the interface. If only one edge of the fine patch falls onto the interface between a regular grid and a subgrid, only one $c$ coefficient is present. If edges 2 and 3 are not on the interface but other edges, (7.32) can be simply permuted.

In addition, the magentic field at this subgrid patch may also be used to obtain electric fields elsewhere not belonging to this patch. This can happen when the coarse patch has two sides or more having subgrids. In this case, (7.32) will have more than 4 entries, whose value can be readily determined from the interpolation of the pertinent electric field unknown from this patch's magnetic field. However, regardless of the number of other electric field unknowns generated from this patch's magnetic field, the $[b]_i^T$ is zero corresponding to other electric field unknowns.

**Irregular patch type 3**

This type of patches is a coarse patch without any subgrid edges, i.e., it consists of the regular edges only, as marked by patch 3 in Fig. 7.2(a) and 7.2(b). However, the magnetic field at this patch is used to generate electric fields elsewhere, and hence the resultant $[a]_i$ vector is different from the regular one. This type of patches are those patches that are connected with the subgrids through vertices, in both 2- and 3-D grids.

In this type of patches, the $[b]_i^T$ remains the same as

$$[b]_i^T = \left[-\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}\right] \tag{7.33}$$

since four electric fields along the patch contour produces the magnetic field at the patch center.

The $[a]_i$ vector however, takes the following irregular form

$$[a]_i = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i} \\ \frac{1}{W_i} \\ -\frac{1}{W_i} \\ \frac{c_1}{L_1^f} \\ \vdots \\ \frac{c_k}{L_k^f} \end{bmatrix}, \tag{7.34}$$

where $c_j (j = 1, 2, ...k)$ are interpolation coefficients whose absolute value is between 0 and 1, but can be either positive or negative, $k$ is the number of electric fields that are generated from the magnetic field at this patch center, and $L_j^f$ are the length parameter of the fine patch that has electric field $j$.

We can also have a complete count of the electric field unknowns generated from type-3 $h_i$. Take patch $k$ shown in Fig. 7.3 as an example, it belongs to type 3. All the electric field unknowns along the left half of $BC$ and upper half of $BE$ that have subgrids will have one entry in $[a]_i$ of patch $k$. In 3-D settings, the side of $BC$ becomes two patches (of a coarse patch size) perpendicular to patch $k$ and centering patch $k$. All electric field unknowns along the subgrid edges on the two patches and parallel with $BC$ will be generated from $h_k$. Similarly, the side of $BE$ also becomes two patches perpendicular to patch $k$ and also centering patch $k$. All electric field unknowns along the subgrid edges on the two patches and parallel with $BD$ will be generated from $h_k$. The above can be extended to the rest of three vertices of patch $k$, if through those vertices, patch $k$ is also attached to subgrids.

### 7.4.2 Estimation of Maximum Time Step

Due to the interpolation scheme, the time step estimated from CFL condition can be inaccurate for a mesh involving subgrids. Although the maximum time step can be calculated from the largest eigenvalue of the system matrix, calculating eigenvalues

can be computationally expensive especially when the unknown size is large, thus we should estimate the time step in a more accurate and efficient way. In the proposed method, since each patch produces a rank-1 matrix, we can estimate the norm of the global system matrix $\mathbf{C}$ by analyzing each rank-1 matrix, thus providing an upper bound of the time step can be used in the time marching.

Since the system matrix $\mathbf{C}$ can be represented as $diag\{\frac{1}{\epsilon}\}\mathbf{S}_h diag\{\frac{1}{\mu}\}\mathbf{S}_e$ where each column of $\mathbf{S}_h$ is $[a]$ and each row of $\mathbf{S}_e$ is $[b]^T$, its norm should satisfy

$$\|\mathbf{C}\| \leq \frac{1}{\mu\epsilon}\|\mathbf{S}_h\|\|\mathbf{S}_e\|. \tag{7.35}$$

Any norm should be sufficient to use here, we choose $\|\mathbf{S}_h\|_1$ and $\|\mathbf{S}_e\|_\infty$ for convenience. Since all the rank-1 matrices for regular cells are the same, we only need to analyze the rank-1 matrices corresponding to the patches adjacent to mesh interface to calculate the norm of $\mathbf{C}$ analytically. Since the spectral radius of $\mathbf{C}$ is less than $\|\mathbf{C}\|$, once $\|\mathbf{C}\|$ is calculated, we can estimate $\Delta t$ as either $2/\sqrt{\|\mathbf{C}\|}$ when $\mathbf{C}$ only has non-negative real eigenvalues or $1/\sqrt{\|\mathbf{C}\|}$ when $\mathbf{C}$ supports complex eigenvalues. Next, we will first show what are the rank-1 matrices for regular patch as well as each irregular patch type, and then analyze them one by one. The patch types described in this section are aligned with those shown in Section 7.4. Based on the following analysis, it is clear to see that the rank-1 matrix corresponding to Irregular Patch Type 2 has the largest norm, thus the time step can be estimated by considering this type of patches only.

**Regular patch (in a uniform or non-uniform grid)**

In both 2- and 3-D settings, the patches that are not adjacent to the interface between a regular grid and a subgrid are considered as regular patches. Their corresponding rank-1 matrices are

$$\mathbf{C}_0 = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i} \\ \frac{1}{W_i} \\ -\frac{1}{W_i} \end{bmatrix} \left[ -\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i} \right]. \tag{7.36}$$

Clearly, the norm of $\mathbf{C}_0$ is

$$\|\mathbf{C}_0\| = \left( \frac{2}{L_i} + \frac{2}{W_i} \right)^2. \tag{7.37}$$

If $L_i = W_i$, the norm is simply $16/{L_i}^2$.

In a non-uniform grid, the average width of the two patches sharing the electric field edge is, in general, used for achieving a better accuracy. In this case, $\mathbf{C}_0$ becomes

$$\mathbf{C}_0 = \begin{bmatrix} -\frac{1}{L1_i} \\ \frac{1}{L2_i} \\ \frac{1}{W1_i} \\ -\frac{1}{W2_i} \end{bmatrix} \left[ -\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i} \right], \tag{7.38}$$

where the length parameters $L1_i, L2_i, W1_i, W2_i$ are averaged between two patches sharing the electric field edge. The norm of $\mathbf{C}_0$ should also be calculated accordingly.

**Irregular patch type 1**

For every patch of this type, the corresponding rank-1 matrix has the following form

$$\mathbf{C}_1 = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i} \\ \frac{1}{W_i} \\ -u \end{bmatrix} \left[ -\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}\tilde{v}^T \right] \tag{7.39}$$

in which $\tilde{v}$ is the $v$ shown in (7.26) extended to length $k$ by appending zeros at the end if $k > n$. For this rank-1 matrix, we can calculate its norm as

$$\|\mathbf{C}_1\| = \left(\frac{2}{L_i} + \frac{1}{W_i} + \sum_{i=1}^{n} |u_i|\right)\left(\frac{2}{L_i} + \frac{1}{W_i} + \frac{1}{W_i}\sum_{i=1}^{k} |v_i|\right). \tag{7.40}$$

**Irregular patch type 2**

For this type of patches, the corresponding rank-1 matrix has the following form

$$\mathbf{C}_2 = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i}(1-c_2) \\ \frac{1}{W_i}(1-c_3) \\ -\frac{1}{W_i} \end{bmatrix} \left[-\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}\right] \tag{7.41}$$

where $c_2$ and $c_3$ both are nonzero, or one of them is zero. When they are nonzero, they are positive coefficients between 0 and 1. Again, the norm of $\mathbf{C}_2$ is

$$\|\mathbf{C}_2\| = \left(\frac{2-c_2}{L_i} + \frac{2-c_3}{W_i}\right)\left(\frac{2}{L_i} + \frac{2}{W_i}\right). \tag{7.42}$$

**Irregular patch type 3**

For every patch of this kind, the corresponding rank-1 matrix has the following form

$$\mathbf{C}_3 = \begin{bmatrix} -\frac{1}{L_i} \\ \frac{1}{L_i} \\ \frac{1}{W_i} \\ -\frac{1}{W_i} \\ \frac{c_1}{L_1^f} \\ \vdots \\ \frac{c_k}{L_k^f} \end{bmatrix} \left[-\frac{1}{L_i}, \frac{1}{L_i}, \frac{1}{W_i}, -\frac{1}{W_i}, zeros(1,k)\right], \tag{7.43}$$

where $c_j$ $(j = 1, \ldots, k)$ are interpolation coefficients that can be either positive or negative, and $k$ zeros are appended at the end of the row vector. The norm of $\mathbf{C}_3$ can be calculated as

$$\|\mathbf{C}_3\| = \left(\frac{2}{L_i} + \frac{2}{W_i} + \sum_{i=1}^{k} \frac{c_k}{L_k^f}\right)\left(\frac{2}{L_i} + \frac{2}{W_i}\right). \tag{7.44}$$

## 7.5 Explicit FDTD Subgridding Algorithm with Unconditional Stability

In existing FDTD subgridding algorithms, temporal subgridding schemes have also been developed to take advantage of the large time step size permitted by the coarse grid, and localize the use of small time step in the subgrid region. In this work, we will leverage our prior work in [66] to make the entire scheme unconditionally stable, while still being explicit. In other words, one can use a large time step size for both regular and subgrid regions.

In the section above, we show that if all the eigenvalues of $\mathbf{C}$ are non-negative real, its explicit time marching is guaranteed to be stable if the time step $\Delta t$ is chosen to be less than $\frac{2}{\sqrt{\lambda_{max}}}$. If $\mathbf{C}$ has complex eigenvalues, then the time step should satisfy $\frac{1}{\sqrt{\lambda_{max}}}$ to guarantee stability. The $\lambda_{max}$ is determined by the smallest space step, and thereby in the subgrid region. On the other hand, given any input pulse of maximum frequency $f_{max}$, a time step less than $\frac{1}{10 f_{max}}$ is sufficient for accuracy. In a subgridding mesh, if the coarse grid size is chosen based on accuracy requirements, the time step required by stability can be estimated as the time step required by accuracy divided by contrast ratio $CR$. When $CR$ is large, the time step required by stability is much smaller than that required by accuracy. To tackle this problem, one can separate the unknowns in the coarse grid from those in subgrids, and solve them in an explicit-implicit fashion. One can also resort to temporal subgridding schemes. Here, we provide an approach based on [66], where the source of instability is found from the fine region and deducted from the system matrix. As a result, an explicit FDTD subgridding algorithm can also be made unconditionally stable. This permits the use

of a large time step size, solely determined by accuracy regardless of space step, in both regular and subgrid regions.

Given any desired time step $\Delta t$, we first categorize all the cells in the grid into two groups. One group $\mathbb{G}_c$ has regular cell sizes and permits the use of the desired time step, while the other group $\mathbb{G}_f$ includes all the fine cells in the subgrids and their adjacent cells that require a smaller time step for a stable simulation. Accordingly, $\mathbf{C}$ can be split into the following two components

$$\mathbf{C} = \mathbf{C}_f + \mathbf{C}_c, \tag{7.45}$$

where $\mathbf{C}_f$ is assembled from $\mathbb{G}_f$, and $\mathbf{C}_c$ is from $\mathbb{G}_c$. Based on (7.12), the $\mathbf{C}_f$ can be obtained by summing up the rank-1 matrix over all the patches in $\mathbb{G}_f$, and hence being

$$\mathbf{C}_f = diag\{\frac{1}{\epsilon}\} \sum_{i=1, i\in\mathbb{G}_f}^{p} \frac{1}{\mu_i}\{a\}_i\{b\}_i^T, \tag{7.46}$$

in which $p$ is the number of patches in the group $\mathbb{G}_f$.

Let the $\mathbf{E}$ and $\mathbf{H}$ unknown number in $\mathbb{G}_f$ be $q$, and $p$ respectively. If we eliminate the zero rows of $\{a\}_i$ and zero columns of $\{b\}_i^T$, (7.46) becomes a small $q$ by $q$ matrix, which can be written as

$$\mathbf{C}_f^{(f)}{}_{q\times q} = \mathbf{A}_{q\times p}\mathbf{B}_{p\times q}^T, \tag{7.47}$$

where $\mathbf{A}$ stores all the $p$ column vectors, and $\mathbf{B}^T$ consists of all the row vectors. We then find the largest $l$ eigenvalues $\lambda_i$ and their corresponding eigenvectors $\mathbf{F}_{hi}^{(f)}$ of $\mathbf{C}_f$ by using Arnorldi method. The complexity of doing so is only $O(l^2q)$. To check whether $\mathbf{F}_{hi}^{(f)}$ are accurate approximations of the original eigenvectors of $\mathbf{C}$, we extend $\mathbf{F}_{hi}^{(f)}$ to $\mathbf{F}_{hi}$ of length $N_e$ based on global unknown ordering. We then perform the following accuracy check:

$$\frac{\|\mathbf{C}\mathbf{F}_{hi} - \lambda_i\mathbf{F}_{hi}\|}{\|\mathbf{C}\mathbf{F}_{hi}\|} < \epsilon. \tag{7.48}$$

Those $\mathbf{F}_{hi}$ satisfying the above accuracy requirement are then identified as the unstable modes. They are first orthogonalized to be $\mathbf{V}_h$, and then deducted from the system matrix as the following

$$\mathbf{C}_l = \mathbf{C} - \mathbf{V}_h\mathbf{V}_h^H\mathbf{C}. \tag{7.49}$$

Table 7.1
Simulation parameters for 2-D wave propagation problem with different contrast ratios

| Contrast 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 |
| | Subgridding | 68 | 116 | 276 | 20256 |
| Time (s) | FDTD | 0.04 | 0.25 | 1.65 | 3418.71 |
| | Subgridding | 0.02 | 0.07 | 0.20 | 96.96 |
| Speedup | | 2 | 3.57 | 8.25 | 35.26 |

The above allows for a much larger time step than $\mathbf{C}$. We then perform an explicit marching on the updated system matrix as

$$\{e\}^{n+1} = 2\{e\}^n - \{e\}^{n-1} - \Delta t^2 \mathbf{C}_l \{e\}^n + \Delta t^2 \{f\}^n \tag{7.50}$$

followed by the following treatment to ensure the resultant $\{e\}$ has no component in $\mathbf{V}_h$ space

$$\{e\}^{n+1} = \{e\}^{n+1} - \mathbf{V}_h \mathbf{V}_h^H \{e\}^{n+1}. \tag{7.51}$$

If $\mathbf{C}$ has complex eigenvalues, we would replace $\mathbf{C}$ in (7.23) by $\mathbf{C}_l$. (7.51) should still be added at each time step.

Since the contribution of $\mathbf{V}_h$ is removed from $\mathbf{C}$, the time marching of (7.50) is stable for the desired large time step. When the time step is chosen based on accuracy, the removed $\mathbf{V}_h$ modes are not required for accuracy either, and hence ensuring accuracy [17, 18, 66].

## 7.6 Numerical Results

In this section, we simulate a variety of 2- and 3-D examples involving different subgrids to demonstrate the validity and efficiency of the proposed method.

### 7.6.1 2-D Wave Propagation

We first simulate a wave propagation problem in a 2-D rectangular region. The grid is shown in Fig. 7.4. Along both $x$- and $y$-axis, the coarse grid size is $L_c = 0.1$ m. To examine the validity of the proposed FDTD subgridding method, the blue region is subdivided into fine grids where the fine grid size is controlled by contrast ratio $CR = \Delta L_c/\Delta L_f$. Fig. 7.4 shows the mesh details when $CR = 5$. The incident electric field is $\mathbf{E}^{inc} = \hat{y}2(t - t_0 - x/c)e^{-(t-t_0-x/c)^2/\tau^2}$ with $c = 3 \times 10^8$ m/s, $\tau = 2 \times 10^{-8}$ s and $t_0 = 4\tau$. All the boundary unknowns are terminated by exact absorbing boundary condition. To check the accuracy of the proposed FDTD subgridding method when $CR = 2$, we first sample the electric field at two observation points located at (0.1, 0.05) m and (0.275, 0.3) m and plot it in Fig. 7.5(a). Point 1 is inside the coarse mesh while point 2 is on the boundary of the subgridding region. The reference result we use here is the analytical solution. For example, the analytical electric field at point $\mathbf{r}_i$ along the direction $\hat{t}_i$ should be $\mathbf{E}_{inc}(\mathbf{r}_i)\cdot\hat{t}_i$. It's clear to see that the simulated fields agree with analytical solution very well. To examine the solution error in the entire computational domain, we calculate the relative error of the entire E unknown vector as $\|\{e\} - \{e\}_{anal}\|/\|\{e\}_{anal}\|$ at each time step with contrast ratio being 2, 5, 10, and 100 respectively. The entire solution error is shown in Fig. 7.5(b). Obviously, the solution accuracy in the entire computational domain is always very good for the four contrast ratios. The lower the contrast ratio, the better the accuracy. Meanwhile, the accuracy is saturated once the contrast ratio reaches a certain value.

To demonstrate the efficiency of the proposed FDTD subgridding method, we also simulate the same problem using the conventional FDTD method with uniform fine grids. The simulation parameters are summarized in Table **??**. As the contrast ratio increases, the largest time step permitted by both the proposed FDTD subgridding method and the conventional FDTD method decreases, while the number of E unknowns increases. Given a contrast ratio, although the proposed FDTD subgridding method has to use the same time step as the conventional FDTD, it can still achieve a

significant CPU time speedup since is has much less unknowns than the conventional FDTD method.

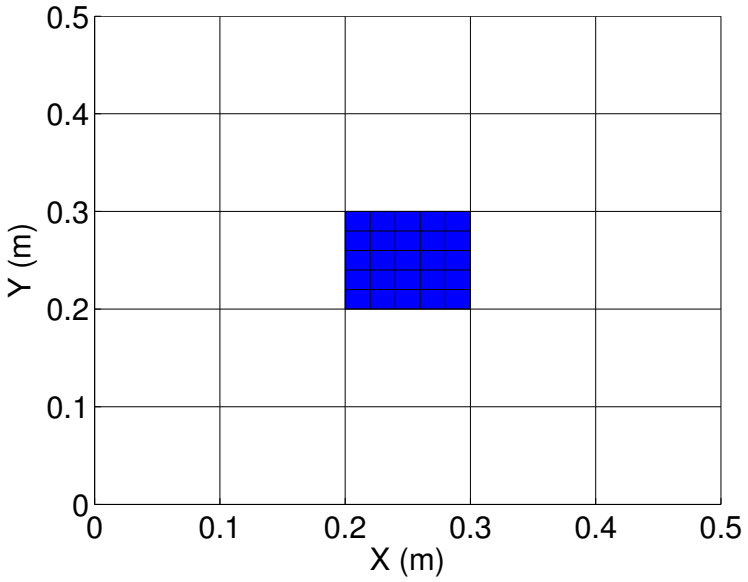
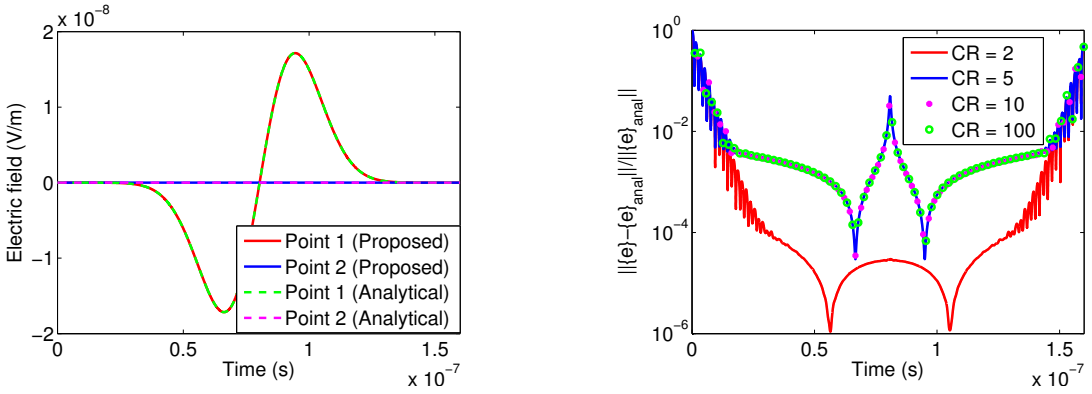



Fig. 7.4. Simulation of a 2-D wave propagation problem: Mesh details.




(a) (b)

Fig. 7.5. Simulation of a 2-D wave propagation problem: (a) Simulated electric field at two observation points in comparison with reference analytical solutions. (b) Entire solution error v.s. time for different contrast ratios.

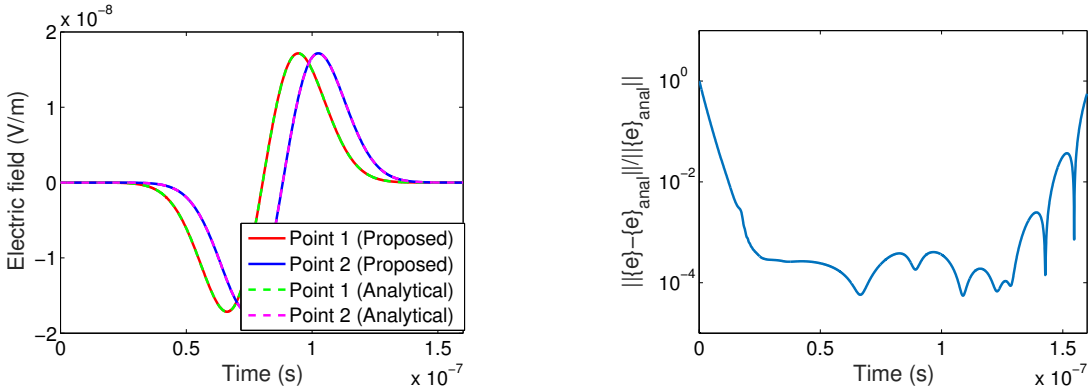
### 7.6.2 3-D Wave Propagation

The second example is a free-space wave propagation problem in a 3-D cube. The size of the computational domain in each direction is 5.1 m. Along all directions, the coarse space step $L_c$ is 0.1 m, resulting in 132,651 coarse cells. The coarse cell at the center is further subdivided into fine cells with contrast ratio being 5, therefore the fine grid size $L_f$ is 0.02 m. The total number of E unknowns in the mesh is 414,240. The same incident field is supplied as that of the first example. Exact absorbing boundary condition is also supplied to all the unknowns on the boundary.

The existence of fine cells renders the time step of the proposed FDTD subgridding method less than $4.0 \times 10^{-11}$ s. Since the analytical solution to this problem is known, we first plot the simulated electric field at two observation points in comparison with analytical solution in Fig. 7.6(a). Point 1 is at (0.1, 0.1, 0.05) m and it's inside the coarse mesh. The location of point 2 is (2.5, 2.5, 2.59) m and it is within the subgridding mesh. Obviously, the electric field waveforms at both points agree with the reference results very well. To examine the solution accuracy in the entire computational domain, in Fig. 7.6(b) we assess the entire solution error measured by $\|\{e\}-\{e\}_{anal}\|/\|\{e\}_{anal}\|$, where $\{e\}$ consists of all 414,240 **E** unknowns obtained from the proposed FDTD subgridding method, while $\{e\}_{anal}$ 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 larger errors at early and late time are because the denominator of the solution error is zero at those times. The proposed FDTD subgridding method takes 201.09 s to finish the simulation. To demonstrate the efficiency of the proposed method, we also discretize the same computational domain into uniform fine grids and simulate the same wave propagation problem in this domain using conventional FDTD method. This uniform fine mesh involves 50,135,040 E unknowns. The times step is the same as that used in the proposed FDTD subgridding method and it takes the conventional FDTD method 29012.74 s. Therefore, the proposed FDTD subgridding method is much faster than the conventional FDTD

method when fine features exist. This is because the number of unknowns is reduced significantly.

We also simulated this example by using the proposed unconditionally stable FDTD subgridding method. First of all, the fine cells are identified, which involves 672 E unknowns and 552 H unknowns, then 320 unstable eigenmodes are extracted from the $\mathbf{C}_f$ assembled from fine cells only. After the contribution of unstable eigenmodes is removed from the system matrix, we are allowed to use $\Delta t = 1.9 \times 10^{-10}$ s that is solely determined by accuracy for time marching. The entire solution error compared to analytical solution at each time step is plotted in Fig. 7.7. It is evident that the accuracy is preserved by comparing Fig. 7.6(b) with Fig. 7.7. Since the proposed FDTD subgridding method can use a much larger time step after the unconditionally stable method is applied, it only takes 28.37 s to finish the simulation.




(a) (b)

Fig. 7.6. Simulation of a 3-D wave propagation problem: (a) Simulated electric field at two observation points in comparison with reference analytical solutions. (b) Entire solution error v.s. time.

### 7.6.3 3-D Cavity with Current Probe Excitation

In this example, we simulate a 3-D cavity excited by a current source as shown in Fig. 7.8(a). The cavity is 1 cm long in all directions and its six faces are all

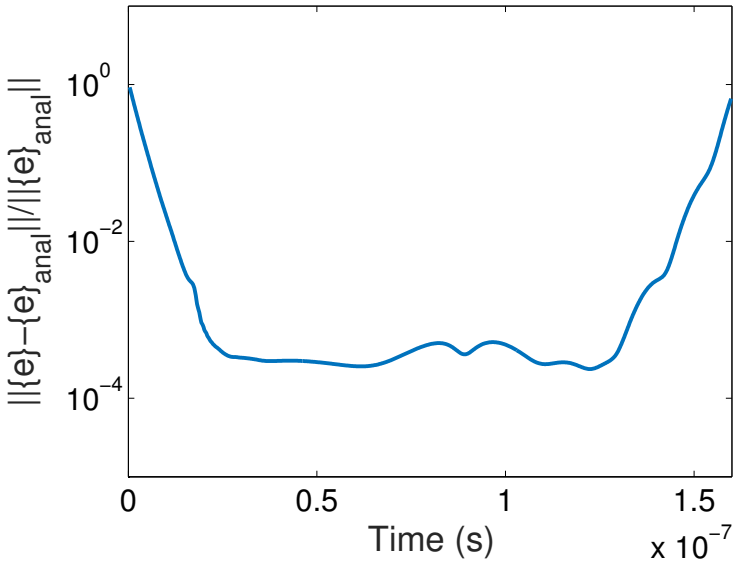
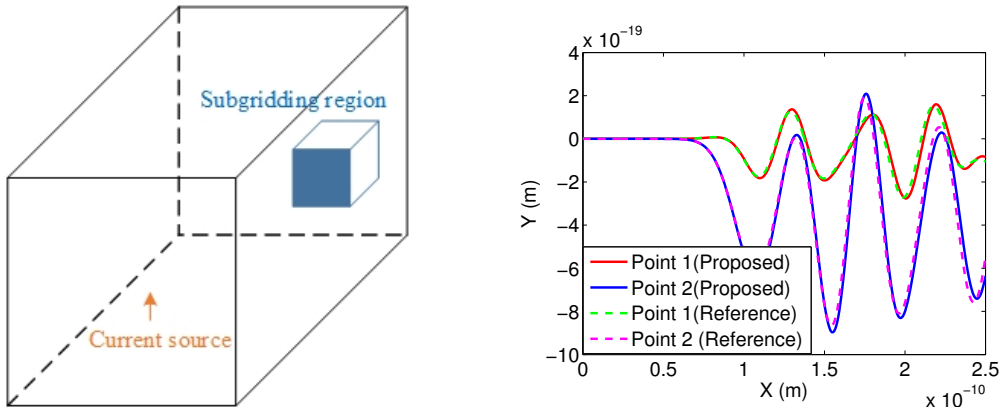



Fig. 7.7. Entire solution error v.s. time when the unconditionally stable methods is applied to the proposed FDTD subgridding method.

terminated by PEC boundary condition. The coarse grid size along each direction is 1 mm except for the blue cube inside the cavity. The blue cube is centered at (4.5, 4.5, 4.5) mm and 1 mm long in all directions. It is filled with conductive material whose conductivity is $5.7 \times 10^7$ S/m. The blue cube is further subdivided into fine mesh whose grid size is 0.2 mm, therefore the contrast ratio $CR$ for this problem is 5. Such a subgridding mesh results in 4,158 E unknowns. A current probe is excited at (2, 2, 1.5) mm. The current is a Gaussian pulse whose waveform is $\mathbf{I} = \hat{z} \exp -(t - t_0)^2/\tau^2$ with $\tau = 2 \times 10^{-11}$ s and $t_0 = 4\tau$. As the reference, we also simulate the same problem using conventional FDTD method with a uniform fine mesh. The total number of E unknowns in this uniform fine mesh is 390,150. Since both the subgridding mesh and the uniform fine mesh have fine grids, the proposed FDTD subgridding method should use the same time step as the conventional FDTD method which is $\Delta t = 3.8 \times 10^{-13}$ s. In Fig. 7.8(b), the electric field sampled at point 1 (8, 8, 7.5) mm and point 2 (4, 4, 9.5) mm is plotted in comparison with reference solution. Overall, the accuracy of the sampled E field is very good. As for the CPU time, the proposed FDTD subgridding

method only takes 0.13 s to finish the entire simulation, while the conventional FDTD method requires 38.68 s, thus a significant speedup is achieved.




(a) (b)

Fig. 7.8. Simulation of a 3-D cavity excited by a current source: (a) Structure details. (b) Simulated electric field at two observation points in comparison with reference FDTD solutions.

### 7.6.4 Inhomogeneous 3-D Phantom Head Beside A Wire Antenna

The last example we study is a large-scale phantom head [57] 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. 7.9. All the boundaries are truncated by PMC. The wire antenna is located at (3.52, 3.52, 2.52) cm, the current on which has a pulse waveform of $\mathbf{I} = 2(t - t_0)e^{-(t-t_0)^2/\tau^2}$ with $\tau = 5.0 \times 10^{-10}$ s and $t_0 = 4\tau$. The coarse step size along $x$-, $y$-, $z$-direction is 4.4 mm, 4.4 mm and 5.6 mm respectively. To capture fine tissues, two coarse cells at the center are subdivided into fine cells in all directions with contrast ratio $CR = 4$, meaning the fine grid size along $x$-, $y$-, $z$-direction is 1.1 mm, 1.1 mm and 1.4 mm respectively. As a result, the total number of E unknowns in this subgridding mesh is 410,300. In conventional FDTD, if fine grids are used everywhere, it would result in 25,428,608 E unknowns. Due to the existence of fine

grids, both the proposed FDTD subgridding method and conventional FDTD method have to use a time step less than $2.2 \times 10^{-12}$ s to ensure stability. In Fig. 7.10(a), the electric field at two observation points whose locations are (3.52, 3.52, 15.96) cm and (24.64, 3.52, 15.96) cm is plotted in comparison with reference solution that is obtained by simulating the same problem in a uniform fine grid using conventional FDTD method. It is clear that the result from the proposed method agrees with the reference result. Since the conventional FDTD method requires a uniform find grid which has much more E unknowns than the proposed FDTD subgridding method, the conventional FDTD method takes 19222.16 s to finish the simulation.

The proposed unconditionally stable FDTD subgridding method is also used to simulate this example. To do so, the fine cells are first identified, which involve 724 electric field unknowns and 594 magnetic field unknowns. Given $\epsilon = 10^{-2}$, 325 unstable eigenmodes are obtained accurately from $\mathbf{S}_f$. With the unstable eigenmodes removed, the largest time step that can be used is increased from $2.2 \times 10^{-12}$ s to $8.8 \times 10^{-12}$ s, which is also the time step solely determined by accuracy. As a result, the unconditionally stable FDTD subgridding method only takes 159.23 s including the time for extracting unstable eigenmodes and explicit time marching. However, without the unconditionally stable method, the FDTD subgridding method needs 528.53 s to finish the same simulation. Therefore, the CPU speedup is 3.32. At each time step, if we denote the solution of all E unknowns obtained from the FDTD subgridding method as $\{e\}_{ref}$, while letting $\{e\}$ be the solution obtained from the unconditionally stable FDTD subgridding method, then we can calculate the relative error as $\|\{e\} - \{e\}_{ref}\|/\|\{e\}_{ref}\|$. In Fig. 7.10(b), the relative error is plotted for the time window when the field solution is nonzero. Obviously, in addition to higher efficiency, the unconditionally stable FDTD subgridding method can also guarantee accuracy across the entire time window.

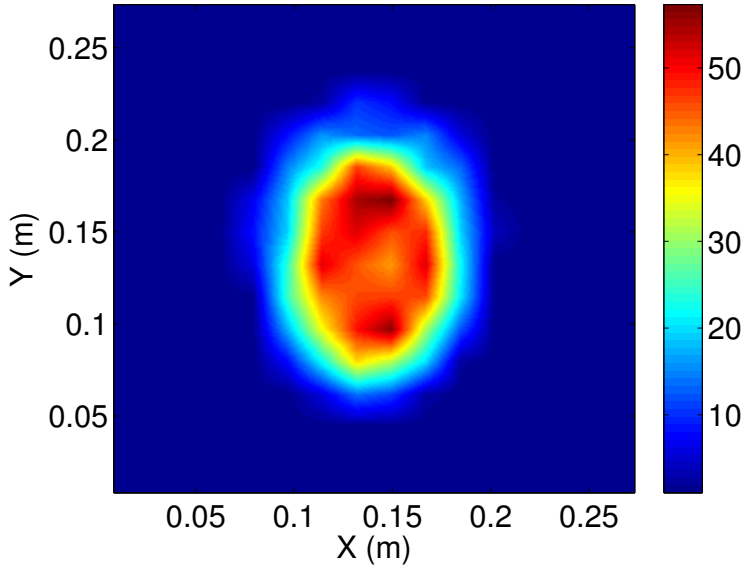
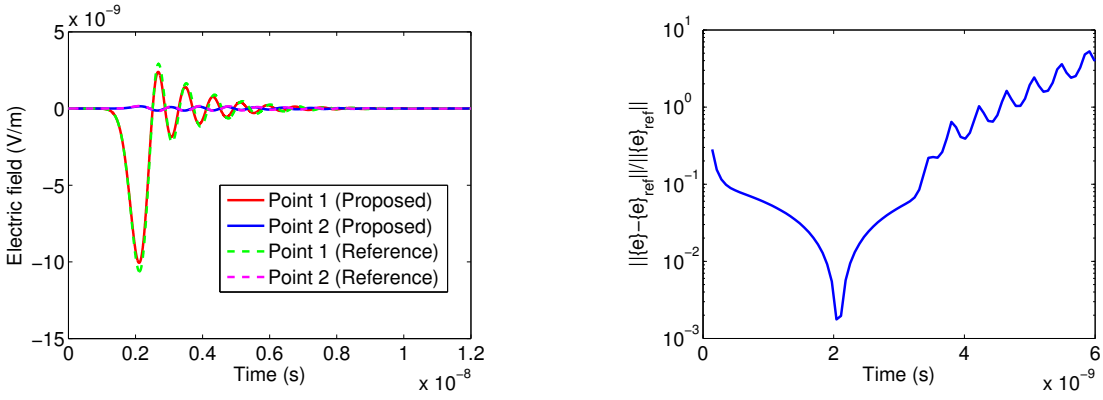



Fig. 7.9. Relative permittivity distribution in a cross section of the phantom head at $z = 2.8$ cm.




(a) (b)

Fig. 7.10. Simulation of a phantom head beside a wire antenna: (a) Simulated electric field at two observation points in comparison with reference FDTD solutions. (b) Entire solution error v.s. time when unconditionally stable method is applied to the proposed FDTD subgridding method.

## 7.7 Conclusion

In this chapter, a novel unsymmetrical but stable FDTD subgridding algorithm is developed for general electromagnetic analysis. We provide a theoretical analysis

to show that an explicit FDTD time marching can be made stable if and only if all the eigenvalues of the governing system matrix are non-negative real. This is satisfied by the original FDTD in a regular grid. However, in a subgridding algorithm, the original FDTD discretization of the curl operators has to be changed to ensure accuracy for field unknowns involved in the subgridding. This change usually results in an unsymmetrical system matrix supporting complex eigenvalues, thus the resulting explicit FDTD time marching becomes definitely unstable. Such an instability may not be observed at early time, but will appear at late time. To resolve this problem, we propose a new time marching scheme to stably simulate the unsymmetrical system, in which the system matrix has an explicit matrix inversion. As a result, the solution can also be obtained explicitly without running into the stability problem. This new time marching scheme provides a flexibility to develop interpolation schemes solely based on accuracy without concerning about the stability. It is also general for use, applicable to other subgridding algorithms. Essentially, this new scheme provides an effective means to explicitly simulate an unsymmetrical numerical system with guaranteed stability.

In addition, in the proposed work, an accurate subgridding algorithm is developed to generate the field unknowns on the subgrid interfaces for both 2-D and 3-D grids. The algorithm allows for an arbitrary grid contrast ratio. The time step allowed for an explicit time marching can be analytically found by analyzing the rank-1 matrices corresponding to the patches adjacent to the subgrid interface. This subgridding algorithm is then further made unconditionally stable. Extensive numerical experiments involving both 2- and 3-D subgrids with various contrast ratios have demonstrated the accuracy, stability, and efficiency of the proposed general method, and new subgridding algorithm.

# 8. MATRIX-FREE TIME-DOMAIN METHOD FOR THERMAL ANALYSIS

## 8.1 Introduction

In previous chapters, we first develop a matrix-free time-domain method in unstructured meshes to perform electromagnetic analysis. In this method, the matrix-free property is independent of element shape. Both accuracy and stability are theoretically guaranteed. The implementation is also straightforward. Then, we reveal that the proposed matrix-free time-domain method naturally reduces to the FDTD method in an orthogonal grid. Therefore, we can solve Maxwell's equations without the need for a matrix solution no matter the discretization of a computational domain is a structured grid or unstructured mesh.

Except for Maxwell's equations, many partial differential equations in other disciplines also require a matrix-free solution. Although those equations may not have the same form as Maxwell's equations, the flexible framework of the proposed matrix-free time-domain method allows for an easy extension to them. In this chapter, we demonstrate that the proposed matrix-free time-domain method can be applied to solve the thermal diffusion equation. Numerical experiments are conducted to show the validity of the proposed method.

## 8.2 Proposed Method

In thermal analysis, we solve the following thermal diffusion equation

$$\tilde{\rho} c_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = P_{joule} + P_0, \tag{8.1}$$

where $k$ is the thermal conductivity, $c_p$ is the specific heat capacity and $\tilde{\rho}$ is the mass density of the material, $T$ is the temperature, and $P_{joule}$ denotes the heat source with

$$P_{joule} = \mathbf{J} \cdot \mathbf{E} = \sigma E^2, \tag{8.2}$$

and $P_0$ represents other heat sources. The conductivity $\sigma$ is a function of temperature

$$\sigma = f(T), \tag{8.3}$$

where function $f$ is material dependent.

In time domain, (8.1) can be solved using finite difference method [67], finite element method [68], finite volume method [69] and many others. Among them, only the finite difference method can be matrix-free in an orthogonal grid. Therefore, the computation can be made much more efficient if the proposed matrix-free time-domain method can be applied to solve (8.1) in unstructured meshes. However, since the matrix-free time-domain method works on two vector variables while (8.1) is a scalar equation, it requires a transformation before the matrix-free time-domain method can be applied.

Given an arbitrary discretization, we can assign the temperature $T$ unknown to the center of every patch, and then attach a direction $\hat{n}$ to it. Thus we can define $\mathbf{T} = T\hat{n}$ where $\hat{n}$ is the unit vector normal to the patch. We also introduce an auxiliary vector $\mathbf{T}_c$ which corresponds to the curl of the $\mathbf{T}$ vector. With these two vector variables, we can cast the original thermal diffusion equation (8.1) into the following two vector equations to solve

$$k\nabla \times \mathbf{T} = -\frac{\partial \mathbf{T}_c}{\partial t}, \tag{8.4}$$

$$\nabla \times \mathbf{T}_c = \tilde{\rho} c_p \mathbf{T} + \mathbf{P}_i. \tag{8.5}$$

To show the equivalency between (8.1) and the above two equations, we consider the source-free scenario. From (8.5), we have

$$\nabla \cdot (\nabla \times \mathbf{T}_c) = \tilde{\rho} c_p \nabla \cdot \mathbf{T} = 0. \tag{8.6}$$

Taking another curl of the left-hand side of (8.4), we should obtain

$$\nabla \times \nabla \times \mathbf{T} = \nabla (\nabla \cdot \mathbf{T}) - \nabla^2 \mathbf{T} = -\nabla^2 \mathbf{T}. \tag{8.7}$$

By eliminating the $\mathbf{T}_c$ unknown from (8.4) and (8.5), we can obtain

$$\tilde{\rho} c_p \frac{\partial \mathbf{T}}{\partial t} + k \nabla \times \nabla \times \mathbf{T} = 0. \tag{8.8}$$

The equation above can be simplified to be the same as (8.1) by utilizing the relationship (8.7). As a result, solving the two vector equations (8.4) and (8.5) simultaneously is equivalent to solving (8.1).

Obviously, (8.4) has the same form as Faraday's law, while (8.5) has the same form as Ampere's law. Hence, the matrix-free time-domain method can be applied to solve (8.4) and (8.5) without any need for solving a matrix equation. First, we can expand $\mathbf{T}_c$ on a set of first-order vector bases, then evaluate (8.5) at $\mathbf{r}_{ti}$ along direction $\hat{h}_{ti}$ $(i = 1, 2, \ldots, N_t)$. Therefore, (8.5) can be discretized as

$$\mathbf{S}_e\{T_c\} = diag\{\tilde{\rho} c_p\}\{T\}. \tag{8.9}$$

On the other hand, by choosing the appropriate $\mathbf{T}$-points located at $\mathbf{r}_{ti}$ and pointing at $\hat{h}_{ti}$ $(i = 1, 2, \ldots, N_t)$, we can disretize (8.4) as

$$diag\{k\}\mathbf{S}_h\{T\} = -\frac{\partial\{T_c\}}{\partial t}. \tag{8.10}$$

In (8.9) and (8.10), both $\mathbf{S}_e$ and $\mathbf{S}_h^T$ are sparse. Their sizes are $N_t \times N_c$ where $N_t$ is the number of $\mathbf{T}$ unknowns while $N_c$ is the number of $\mathbf{T}_c$ unknowns. $diag\{k\}$ and $diag\{\tilde{\rho} c_p\}$ are diagonal matrices with diagonal entries being $k_i$ and $\tilde{\rho}_i c_{pi}$ respectively. Vector $\{T\}$ contains all the $\mathbf{T}$ unknowns, while vector $\{T_c\}$ contains all the $\mathbf{T}_c$ unknowns.

(8.9) and (8.10) can be solved without any matrix solution using forward difference scheme. Alternatively, we can also eliminate $\mathbf{T}$ unknowns and solve for $\mathbf{T}_c$ unknowns first as follows

$$\frac{\partial\{T_c\}}{\partial t} + \mathbf{S}\{T_c\} = 0, \tag{8.11}$$

where

$$\mathbf{S} = diag\{\frac{k}{\tilde{\rho}c_p}\}\mathbf{S}_h\mathbf{S}_e. \tag{8.12}$$

(8.11) can be discretized in time using forward difference scheme as follows

$$\{T_c\}^{n+1} = \{T_c\}^n - \Delta t\mathbf{S}\{T_c\}^n. \tag{8.13}$$

The stability of (8.13) is guaranteed as long as $\Delta t < \frac{2Re(\lambda_i)}{|\lambda_i|^2}$ where $\lambda_i$ is an arbitrary eigenvalue of $\mathbf{S}$.

## 8.3 Numerical Results

In this section, we simulate a few examples including 2-D and 3-D cases to validate the correctness of the proposed method both in time domain and frequency domain.

### 8.3.1 Copper Plane with Heat Conduction in Orthogonal Grid

In this example, we consider a piece of copper plane with each side being 0.3 m. The copper plane is discretized into uniform orthogonal grid with space step being 0.01 m. The temperature on one side of the plane is 500 $^\circ C$ while 100 $^\circ C$ on other sides. The heat conduction parameters for copper is $k = 398W/(m \cdot K)$, $c_p = 386J/(kg \cdot K)$ and $\tilde{\rho} = 8930kg/m^3$. To guarantee the stability of the time marching scheme in (8.13), the maximum time step allowed is 0.21 s. In Fig. 8.1, we first plot the temperature at point (0.1, 0.1) m as a function of time. Obviously, the temperature is 0 $^\circ C$ in the beginning, then quickly grows until it reaches steady state. This behavior also matches with the physical process of heat conduction. At steady state, the temperature distribution of the copper plane is shown in Fig. 8.2. From [67], it is known that the steady-state temperature at points (0.1, 0.1) m, (0.1, 0.2) m, (0.2, 0.1) m and (0.2, 0.2) m should be 150 $^\circ C$, 150 $^\circ C$, 250 $^\circ C$, 250 $^\circ C$ respectively due to symmetry. In Table 8.1, we list both the reference solution and the simulated result from the proposed method. The absolute error is less than 3 $^\circ C$ for all the observation points, which validates the accuracy of the proposed method.

Table 8.1
The steady-state temperature at observation points

| Point Location | (0.1, 0.1) | (0.1, 0.2) | (0.2, 0.1) | (0.2, 0.2) |
|---|---|---|---|---|
| Reference ($^{\circ}C$) | 150 | 250 | 150 | 250 |
| Orthogonal Grid ($^{\circ}C$) | 147.7473 | 252.2526 | 147.7473 | 252.2526 |
| Unstructured Mesh ($^{\circ}C$) | 154.5642 | 245.4260 | 154.5642 | 245.4260 |

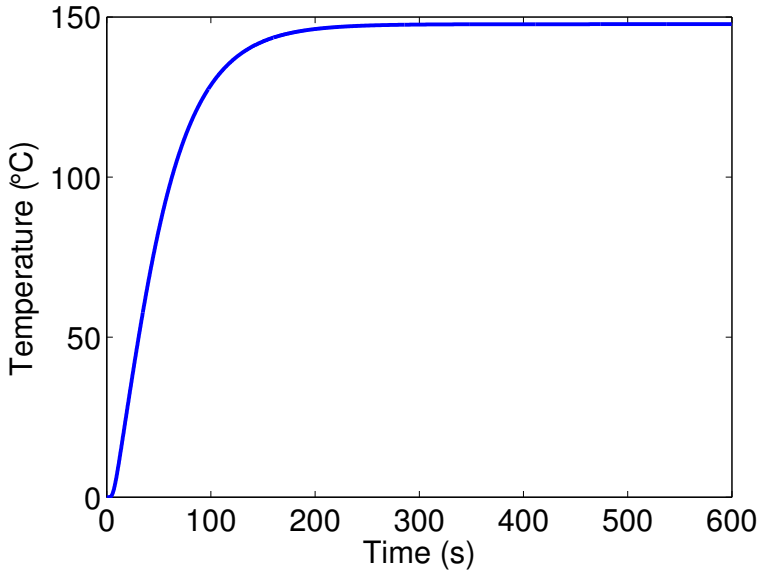



Fig. 8.1. Temperature v.s. time at an observation point.

### 8.3.2 Copper Plane with Heat Conduction in Triangular Mesh

Different from Section 8.3.1, the same copper plane is discretized into an triangular mesh shown in Fig. 8.3. In frequency domain, (8.4) and (8.5) have analytical solutions as follows

$$\mathbf{T} = \hat{z}\sqrt{\frac{\omega}{k\tilde{\rho}c_p}}\left(\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}j\right)e^{jlx}, \tag{8.14}$$

$$\mathbf{T}_c = -\hat{y}e^{jlx}, \tag{8.15}$$

$$l = \sqrt{\frac{\omega\tilde{\rho}c_p}{k}}\left(-\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}j\right). \tag{8.16}$$

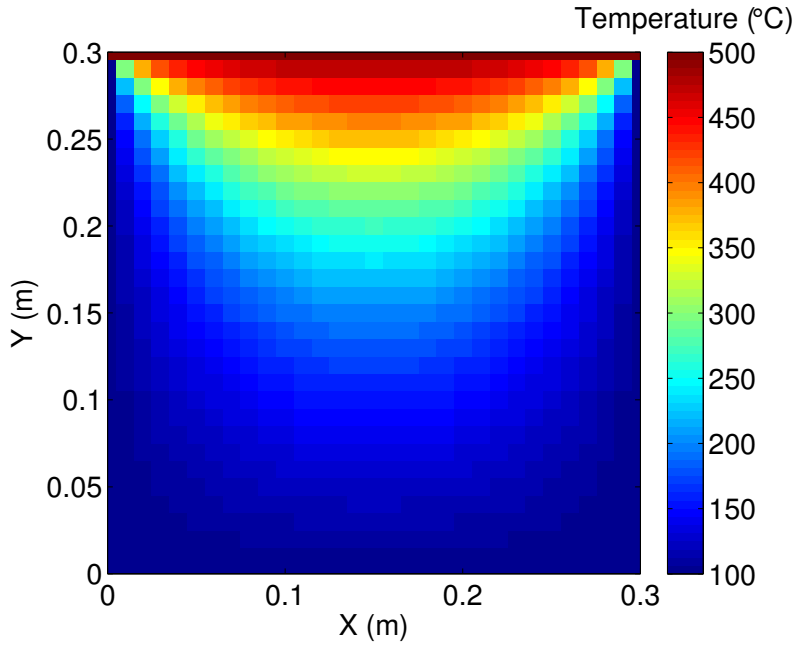



Fig. 8.2. Temperature distribution at steady state.

Table 8.2
The accuracy of $\{T_c\}$ and $\{T\}$ at different frequencies

| Frequency | Relative error of $\{T_c\}$ | Relative error of $\{T\}$ |
|---|---|---|
| $10^{-5}$ | $8.6687 \times 10^{-4}$ | $3.1069 \times 10^{-5}$ |
| $10^{-4}$ | 0.0028 | $3.1223 \times 10^{-4}$ |
| $10^{-3}$ | 0.0107 | 0.0040 |
| $10^{-2}$ | 0.0815 | 0.0833 |

Hence, we can supply an analytical solution to the boundary unknowns, and examine the accuracy of the proposed matrix-free method in the triangular mesh at different frequencies. Table 8.2 shows the relative error of both $\{T_c\}$ and $\{T\}$ at different frequencies. The relative error is calculated as $\|\{T\} - \{T\}_{anal}\|/\|\{T\}_{anal}\|$. It can be seen clearly that the accuracy of the unknowns as compared to the low-frequency analytical solution becomes better when frequency gets lower. Overall, the accuracy is very good at frequencies lower than 0.1 Hz.

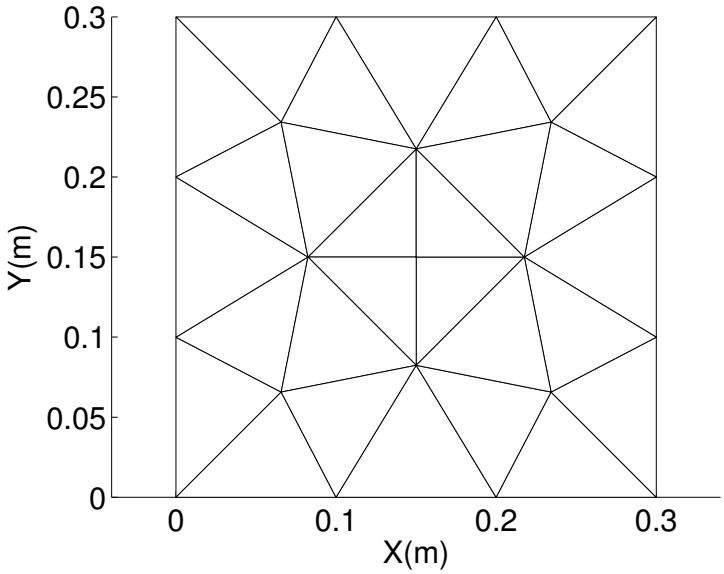



Fig. 8.3. Mesh details of a copper plane.

Similarly, we also supply the same boundary condition as Section 8.3.1 and study how the transient temperature at each point changes. The time step we use here is 1.67 s. In Fig. 8.4, we plot the temperature at point (0.1, 0.1) m as a function of time. It is clear to see that the temperature at this point starts to grow from initial temperature 0 $^{\circ}C$ and then saturates to its steady-state value of 154.5643 $^{\circ}C$. From Section 8.3.1, we know the reference solution at this point at steady state should be 150 $^{\circ}C$, thus the absolute error of the solution at this point is 4.5643 $^{\circ}C$. We also list the steady-state temperature at the same observation points in Table 8.1. The absolute error is less than 5.5 $^{\circ}C$ for all the four points. Meanwhile, we also plot the temperature distribution across the whole plane at steady state in Fig. 8.5. If we consider the result solved using orthogonal grid as a reference, the relative error $\|\{T\} - \{T\}_{ref}\|/\|\{T\}_{ref}\|$, where $\{T\}$ is the solution vector containing the steady-state temperature at all points and $\{T\}_{ref}$ is the reference solution vector, is 6.06%. Therefore, the proposed matrix-free time-domain method can provide an accurate solution to the thermal diffusion equation not only at one point but also in the entire computational domain.

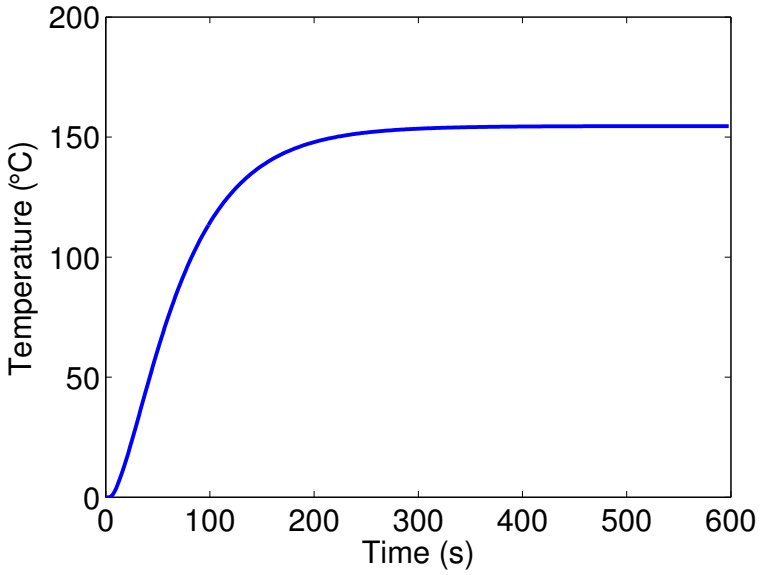



Fig. 8.4. Temperature v.s. time at an observation point.

Since there is no analytical solution to the problem we study above, the solution obtained from all numerical methods has error, thus even the solution solved in a very fine orthogonal grid can not serve as a perfect reference solution to check the accuracy of the proposed method. To examine the accuracy of the proposed method in a fair way, we can supply a homogeneous boundary condition to the copper plane such that the steady-state solution is known. For example, if we set the boundary on each side of the plane to be 100 $^{\circ}C$, the temperature at any point at steady state should also be 100 $^{\circ}C$. Given such a problem, we can obtain the simulated temperature at all points at steady state using the proposed method, then compare it with the analytical reference solution. The relative error is $1.9 \times 10^{-5}$, which validates the accuracy of the proposed method.

### 8.3.3 Copper Cube with Heat Conduction in Tetrahedral Mesh

A cube of size $1 \times 0.5 \times 0.75$ $m^3$ is discretized into tetrahedral mesh. The discretization details are shown in Fig. 8.6. The temperature on every side plane is set

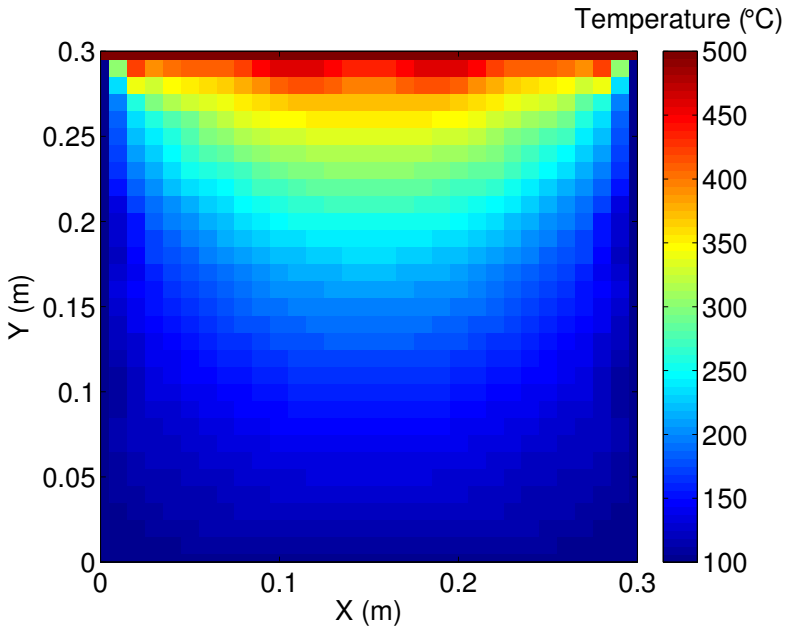



Fig. 8.5. Temperature distribution at steady state.

to be 100 $^{\circ}C$. To guarantee the stability of the proposed method, we choose the time step $\Delta t = 0.9$ s. In Fig. 8.7, the temperature at point (0.4747, 0.2197, 0.6826) m is plotted versus time. Clearly, the temperature at this point gradually grows and finally reaches its steady-state value that is 100 $^{\circ}C$. To examine the solution accuracy of all points, we calculate $\|\{T\} - \{T\}_{ref}\|/\|\{T\}_{ref}\|$ at steady state where $\{T\}_{ref}$ is a vector containing the analytical solution at every point. The relative error is $2.2 \times 10^{-4}$, thus the proposed method has no difficulty in producing accurate results in an unstructured tetrahedral mesh.

## 8.4 Conclusion

In this chapter, we demonstrate that the matrix-free time-domain method developed in previous chapters can be applied to solve thermal diffusion equations in both orthogonal grids and unstructured meshes. To do so, the scalar thermal diffusion equation is transformed to two vector equations to solve. The equivalence between

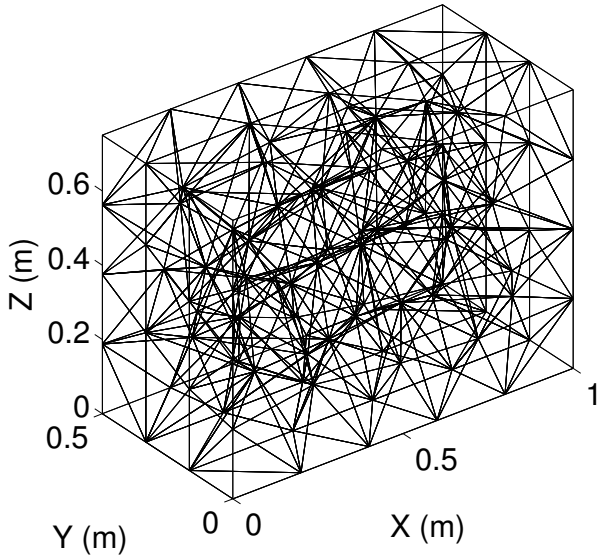



Fig. 8.6. Mesh details of a copper cube.

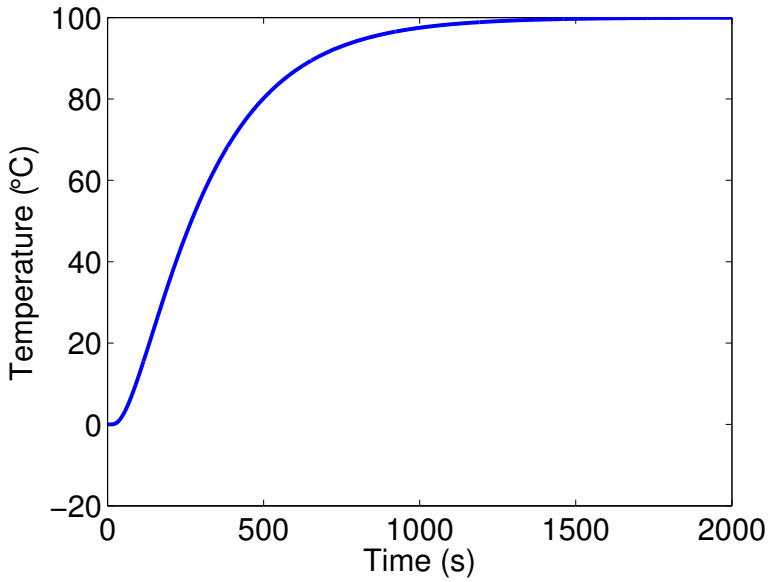



Fig. 8.7. Temperature v.s. time at an observation point.

them is also proved. All the advantages of the matrix-free time-domain method are preserved when solving thermal diffusion equations. Numerical experiments in both

time domain and frequency domain have validated the correctness of the proposed method.

# 9. CONCLUSIONS AND FUTURE WORK

In this work, we develop a new matrix-free time-domain method in arbitrary unstructured meshes to overcome the challenge of solving large-scale problems accurately and efficiently, when arbitrarily shaped geometries and materials are involved. The matrix-free property of the proposed method 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, and it can be applied to solve the partial differential equations in other disciplines. In addition, in a structured grid and with zeroth-order vector bases, the proposed method reduces exactly to the FDTD.

To create the proposed new method, we have considered the following three aspects that are equally important.

- *Matrix-Free Discretization of Maxwell's equations*: In order to create a matrix-free time-domain method, it is desired to make the matrix in front of the resultant second-order time derivative term diagonal. This matrix is, in general, termed as mass matrix. Motivated by this requirement, we pursue a discretization of Faraday's law and Ampere's law such that, at each time step, **H** unknowns can be obtained accurately from **E** unknowns via Faraday's law, then **E** unknowns can be obtained accurately from **H** unknowns through Ampere's law. The first goal is achieved by expanding the electric field in each element by vector basis functions. To achieve the second goal, we propose to sample **H** unknowns across the elements sharing **E** unknown in such a way that they can reversely produce the first field unknown accurately, without any need for inter-

polation and projection either. Obviously, there is no dual mesh involved. We can certainly expand **H** field using vector basis functions while sample **E** field to meet the need of different problems. The new discretization of Maxwell's equations is the key to realize the matrix-free property of the proposed method.

- *Accuracy*: While making the proposed method free of matrix solution, accuracy must be ensured. In Faraday's law, we expand the **E** field by a set of vector basis functions. If zeroth-order vector basis functions are used, the curl of **E** becomes a constant in each element, thus only the **H** field at the element center and perpendicular to the element can be obtained accurately. However, the **H** field has to be sampled in such a way that **E** unknowns can be accurately calculated from those **H** sampled points, which can be located at any point inside the element. Therefore, to resolve this problem, we only need to go one order higher when choosing the vector basis functions for **E**. If so, the **H** field at any point along any direction can be obtained accurately, thus the accuracy is also guaranteed.

- *Time marching stability*: With the previous two essential points addressed, the proposed discretization of Maxwell's equation results in a numerical system free of matrix solution without losing accuracy. However, as a time-domain method, stability also has to be guaranteed. Unlike the explicit FDTD method, the curl operator in Faraday's law is discretized in a different way in the proposed method than that in Ampere's law, which results in an unsymmetrical system matrix. It is proved that the traditional explicit time marching scheme is definitely unstable if the resultant unsymmetrical system matrix supports complex eigenvalues. To guarantee stability, we propose to employ a backward-difference scheme instead of a central-difference scheme to discretize the time derivatives since backward difference scheme can always be stable even though complex eigenvalues exist. One drawback of using the backward difference scheme is that it ruins the matrix-free property. However, this problem is easy to solve

since the mass matrix is diagonal. With the traditional central-difference time step, the inverse of the left-hand-side matrix in the final update equation can be replaced by a series expansion. Therefore, no matrix inversion is involved. Instead, we only need to perform a few matrix-vector multiplications. In such a way, we can guarantee the stability in time while preserving the matrix-free property.

Since the maximum time step allowed by the proposed matrix-free time-domain method is restricted by the minimum space step in the mesh, we also develop a new matrix-free time-domain method with unconditionally stability to break the barrier of time step. Basically, we first find out the root cause of instability and then directly eradicate it from the system matrix. The computation of finding the unstable modes is very cheap since it only requires the calculation of the largest $k$ eigenvalues and their corresponding eigenvectors from a sparse matrix. As a result, the advantages of a matrix-free method in time domain are accentuated, while its shortcoming in time step is remedied, permitting an efficient analysis of large-scale and multi-scale problems.

In addition, the proposed matrix-free methods naturally reduce to the FDTD method in an orthogonal grid, but with a new formulation that is a patch-based and single-grid representation. This formulation reveals that the curl-curl operator has a natural rank-1 decomposition, which permits an efficient extraction of unstable eigenmodes from fine cells only. Based on this finding, we develop a fast explicit and unconditionally stable FDTD method. Using the new patch-based rank-1 formulation, we also develop a new subgridding algorithm which locally refines the mesh at regions requiring a higher resolution to further improve the efficiency of the FDTD method. A theoretical stability analysis is also presented to show that the stability of the proposed subgridding algorithm is guaranteed, although the system matrix is unsymmetric.

The future research potentials of this work include but not limited to

- *Higher-order* **H** *sampling*: Currently, we expand the **E** field by a set of first-order vector basis functions while the **H** field sampling is actually still in its zeroth-order. If we can sample more **H** fields whose locations and directions are related to the first-order vector basis functions in a rectangular loop, we expect the accuracy of the entire solution to be even better. Notice that the number of **E** unknowns remains the same.

- *Property of unsymmetrical system matrix*: Since our current discretization results in an unsymmetrical system matrix **S**, it can support complex eigenvalues or even negative ones. On the other hand, it can also be positive semi-definite, thus having non-negative real eigenvalues only. The property of the unsymmetrical matrix resulting from the matrix-free method will be further studied. In addition, a symmetrical matrix-free operator will also be pursued.

- *Application to realistic problems*: Although many numerical examples have been simulated to demonstrate the generality, efficiency and stability of the proposed matrix-free time-domain method, we still pursue to solve more realistic problems using the proposed method, for example, product-level full package involving different kinds of inhomogeneous and conductive materials. To demonstrate the efficiency, more numerical methods in addition to the traditional finite element method can be considered to compare the CPU time with the proposed method for a given problem, especially in an unstructured mesh.

- *Application to other research areas*: The proposed matrix-free time-domain method provides a flexible framework for solving problems in not only electromagnetics but also many other research areas. The thermal analysis has been conducted using the proposed matrix-free time-domain method in Chap. 8. There also exist many problems in other disciplines that demand an efficient matrix-free solution. For example, the simulation of nano-scale structures, analysis of materials involving dispersion and anisotropy, incorporation of compli-

cated boundary conditions, multiphysics simulations across different disciplines, etc.

# LIST OF REFERENCES

# LIST OF REFERENCES


[1] K. S. Yee *et al.*, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, 1966.

[2] A. Taflove and S. C. Hagness, *Computational electrodynamics*. Artech House Publishers, 2000.

[3] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 10, pp. 2003–2007, 1999.

[4] F. Zheng, Z. Chen, and J. Zhang, "A finite-difference time-domain method without the Courant stability conditions," *IEEE Microwave and Guided Wave Letters*, vol. 9, no. 11, pp. 441–443, 1999.

[5] C. Sun and C. Trueman, "Unconditionally stable Crank-Nicolson scheme for solving two-dimensional Maxwell's equations," *Electronics Letters*, vol. 39, no. 7, pp. 595–597, 2003.

[6] J. Lee and B. Fornberg, "A split step approach for the 3-D Maxwell's equations," *Journal of Computational and Applied Mathematics*, vol. 158, no. 2, pp. 485–505, 2003.

[7] G. Zhao and Q. H. Liu, "The unconditionally stable pseudospectral time-domain (PSTD) method," *IEEE Microwave and Wireless Components Letters*, vol. 13, no. 11, pp. 475–477, 2003.

[8] J. Shibayama, M. Muraki, J. Yamauchi, and H. Nakano, "Efficient implicit FDTD algorithm based on locally one-dimensional scheme," *Electronics Letters*, vol. 41, no. 19, pp. 1046–1047, 2005.

[9] V. E. Do Nascimento, B. Borges, and F. L. Teixeira, "Split-field PML implementations for the unconditionally stable LOD-FDTD method," *IEEE Microwave and Wireless Components Letters*, vol. 16, no. 7, p. 398, 2006.

[10] Y.-S. Chung, T. K. Sarkar, B. H. Jung, and M. Salazar-Palma, "An unconditionally stable scheme for the finite-difference time-domain method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 3, pp. 697–704, 2003.

[11] Z. Chen, Y.-T. Duan, Y.-R. Zhang, and Y. Yi, "A new efficient algorithm for the unconditionally stable 2-D WLP-FDTD method," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 7, pp. 3712–3720, 2013.

[12] Z.-Y. Huang, L.-H. Shi, B. Chen, and Y.-H. Zhou, "A new unconditionally stable scheme for FDTD method using associated Hermite orthogonal functions," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 9, pp. 4804–4809, 2014.

[13] E. L. Tan, "Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 1, pp. 170–177, 2008.

[14] Q. He and D. Jiao, "An explicit time-domain finite-element method that is unconditionally stable," in *IEEE International Symposium on Antennas and Propagation (AP-S)*, pp. 1–4, IEEE, 2011.

[15] C. Chang and C. D. Sarris, "A spatially filtered finite-difference timedomain scheme with controllable stability beyond the CFL limit," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 3, pp. 351–359, 2013.

[16] Q. He and D. Jiao, "Explicit time-domain finite-element method stabilized for an arbitrarily large time step," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 11, pp. 5240–5250, 2012.

[17] M. Gaffar and D. Jiao, "An explicit and unconditionally stable FDTD method for electromagnetic analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 11, pp. 2538–2550, 2014.

[18] M. Gaffar and D. Jiao, "Alternative method for making explicit FDTD unconditionally stable," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 12, pp. 4215–4224, 2015.

[19] 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 Transactions on Microwave Theory and Techniques*, vol. 39, no. 3, pp. 471–479, 1991.

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

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

[22] M. J. White, Z. Yun, and M. F. Iskander, "A new 3D FDTD multigrid technique with dielectric traverse capabilities," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 3, pp. 422–430, 2001.

[23] R. Holland, "Finite-difference solution of Maxwell's equations in generalized nonorthogonal coordinates," *IEEE Transactions on Nuclear Science*, vol. 6, no. 30, pp. 4589–4591, 1983.

[24] M. Fusco, "FDTD algorithm in curvilinear coordinates [EM scattering]," *IEEE Transactions on Antennas and Propagation*, vol. 38, no. 1, pp. 76–89, 1990.

[25] J.-F. Lee, R. Palandech, and R. Mittra, "Modeling three-dimensional discontinuities in waveguides using nonorthogonal FDTD algorithm," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, no. 2, pp. 346–352, 1992.

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

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

[28] Y. Hao and C. J. Railton, "Analyzing electromagnetic structures with curved boundaries on cartesian FDTD meshes," *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, no. 1, pp. 82–88, 1998.

[29] N. K. Madsen, "Divergence preserving discrete surface integral methods for Maxwell's equations using nonorthogonal grids," *Journal of Computational Physics*, vol. 119, pp. 34–45, 1995.

[30] C. Chan, J. Elson, and H. Sangani, "An explicit finite-difference time-domain method using whitney elements," in *IEEE International Symposium on Antennas and Propagation (AP-S)*, vol. 3, pp. 1768–1771, IEEE, 1994.

[31] S. Gedney, F. Lansing, and D. Rascoe, "A full-wave analysis of passive monolithic integrated circuit devices using a generalized Yee-algorithm," *IEEE Transactions on Microwave Theory and Techniques*, vol. 44, no. 8, pp. 1393–1400, 1996.

[32] A. Bossavit and L. Kettunen, "Yee-like schemes on a tetrahedral mesh, with diagonal lumping," *International Journal of Numerical Modeling-Electronic Networks Devices and Fields*, vol. 12, no. 1, pp. 129–142, 1999.

[33] C. Lee, B. McCartin, R. Shin, and J. A. Kong, "A triangle grid finite-difference time-domain method for electromagnetic scattering problems," *Journal of Electromagnetic Waves and Applications*, vol. 8, no. 4, pp. 449–470, 1994.

[34] M. Hano and T. Itoh, "Three-dimensional time-domain method for solving Maxwell's equations based on circumcenters of elements," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 946–949, 1996.

[35] S. D. Gedney and J. A. Roden, "Numerical stability of nonorthogonal FDTD methods," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 2, pp. 231–239, 2000.

[36] M. Cinalli and A. Schiavoni, "A stable and consistent generalization of the FDTD technique to nonorthogonal unstructured grids," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 5, pp. 1503–1512, 2006.

[37] D. Jiao and J.-M. Jin, *The finite element method in electromagnetics*, ch. Finite element analysis in time domain, pp. 529–584. John Wiley & Sons, 2002.

[38] M. Feliziani and F. Maradei, "Hybrid finite-element solutions as time dependent Maxwell's curl equations," *IEEE Transactions on Magnetics*, vol. 31, no. 3, pp. 1330–1335, 1995.

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

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

[41] S. D. Gedney, T. Kramer, C. Luo, J. Roden, R. Crawford, B. Guernsey, J. Beggs, and J. Miller, "The discontinuous Galerkin finite element time domain method (DGFETD)," pp. 1–4, IEEE, 2008.

[42] 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 Transactions on Antennas and Propagation*, vol. 60, no. 4, pp. 1969–1977, 2012.

[43] R. D. Graglia, D. R. Wilton, and A. F. Peterson, "Higher order interpolatory vector bases for computational electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 329–342, 1997.

[44] Q. He, H. Gan, and D. Jiao, "Explicit time-domain finite-element method stabilized for an arbitrarily large time step," *IEEE Transactions on Antennas and Propagation*, vol. 60, no. 11, pp. 5240–5250, 2012.

[45] F. Tisseur and K. Meerbergen, "The quadratic eigenvalue problem," *SIAM Review*, vol. 43, no. 2, pp. 235–286, 2001.

[46] P.-O. Persson and G. Strang, "A simple mesh generator in MATLAB," *SIAM review*, vol. 46, no. 2, pp. 329–345, 2004.

[47] J. Yan and D. Jiao, "A matrix-free time-domain method independent of element shape for electromagnetic analysis," in *IEEE International Symposium on Antennas and Propagation (AP-S)*, pp. 2258–2259, IEEE, 2014.

[48] J. Yan and D. Jiao, "Accurate matrix-free time-domain method in unstructured meshes," in *IEEE International Microwave Symposium (IMS)*, pp. 1–4, IEEE, 2015.

[49] J. Yan and D. Jiao, "Accurate matrix-free time-domain method in three-dimensional unstructured meshes," in *IEEE International Symposium on Antennas and Propagation (AP-S)*, pp. 1830–1831, IEEE, 2015.

[50] J. Yan and D. Jiao, "Accurate and stable matrix-free time-domain method in 3-D unstructured meshes for general electromagnetic analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 12, pp. 4201–4214, 2015.

[51] J. Yan and D. Jiao, "Matrix-free time-domain method for general electromagnetic analysis in 3-D unstructured meshes – modified-basis formulation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 8, pp. 2371–2382, 2016.

[52] J. Jin, *The finite element method in electromagnetics*. John Wiley & Sons, 2014.

[53] M.-F. Wong, O. Picon, and V. F. Hanna, "A finite element method based on whitney forms to solve Maxwell equations in the time domain," *IEEE Transactions on Magnetics*, vol. 31, no. 3, pp. 1618–1621, 1995.

[54] B. Zhou and D. Jiao, "Direct finite-element solver of linear complexity for large-scale 3-d electromagnetic analysis and circuit extraction," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 10, pp. 3066–3080, 2015.

[55] M. Gaffar and D. Jiao, "An explicit and unconditionally stable FDTD method for the analysis of general 3-d lossy problems," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 9, pp. 4003–4015, 2015.

[56] J. Lee, D. Chen, V. Balakrishnan, C.-K. Koh, and D. Jiao, "A quadratic eigenvalue solver of linear complexity for 3-D electromagnetics-based analysis of large-scale integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 3, pp. 380–390, 2012.

[57] I. G. Zubal, C. R. Harrell, E. O. Smith, Z. Rattner, G. Gindi, and P. B. Hoffer, "Computerized three-dimensional segmented human anatomy," *Medical physics*, vol. 21, no. 2, pp. 299–302, 1994.

[58] K. S. Kunz and L. Simpson, "A technique for increasing the resolution of finite-difference solutions of the Maxwell equation," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-23, no. 4, pp. 419–422, 1981.

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

[60] M. Okoniewski, E. Okoniewska, and M. A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 422–429, 1997.

[61] 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 Transactions on Antennas and Propagation*, vol. 55, no. 7, pp. 1981–1990, 2007.

[62] P. Thoma and T. Weiland, "A consistent subgridding scheme for the finite difference time domain method," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 9, no. 5, pp. 359–374, 1996.

[63] O. Podebrad, M. Clemens, and T. Weiland, "New flexible subgridding scheme for the finite integration technique," *IEEE Transactions on Magnetics*, vol. 39, no. 3, pp. 1662–1665, 2003.

[64] N. V. Venkatarayalu, R. Lee, Y.-B. Gan, and L.-W. Li, "A stable FDTD subgridding method based on finite element formulation with hanging variables," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 907–915, 2007.

[65] K. Krishnaiah and C. J. Railton, "A stable subgridding algorithm and its application to eigenvalue problems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 5, pp. 620–628, 1999.

[66] J. Yan and D. Jiao, "Explicit and unconditionally stable FDTD method without eigenvalue solution," in *IEEE International Microwave Symposium*, pp. 1–4, IEEE, 2016.

[67] J. Holman, *Heat transfer*. McGraw-Hill Education, 2009.

[68] T. Lu and J.-M. Jin, "Electrical-thermal co-simulation for DC IR-drop analysis of large-scale power delivery," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 4, no. 2, pp. 323–331, 2014.

[69] J. Xie and M. Swaminathan, "Electrical-thermal co-simulation of 3D integrated systems with micro-fluidic cooling and joule heating effects," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 2, pp. 234–246, 2011.

[70] R. D. Graglia, D. R. Wilton, A. F. Peterson, and I.-L. Gheorma, "Higher order interpolatory vector bases on prism elements," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 3, pp. 442–450, 1998.

# APPENDICES

# A. FIRST-ORDER CURL-CONFORMING VECTOR BASIS FUNCTIONS IN TETRAHEDRAL ELEMENT

In a tetrahedral element, among the 20 first-order vector bases [43], there are 12 edge vector basis functions, which are defined as

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

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

$$\begin{aligned}
\mathbf{W}_1 &= L_1(\xi_2 \nabla\xi_1 - \xi_1 \nabla\xi_2) \\
\mathbf{W}_2 &= L_2(\xi_1 \nabla\xi_3 - \xi_3 \nabla\xi_1) \\
\mathbf{W}_3 &= L_3(\xi_4 \nabla\xi_1 - \xi_1 \nabla\xi_4) \\
\mathbf{W}_4 &= L_4(\xi_3 \nabla\xi_2 - \xi_2 \nabla\xi_3) \\
\mathbf{W}_5 &= L_5(\xi_2 \nabla\xi_4 - \xi_4 \nabla\xi_2) \\
\mathbf{W}_6 &= L_6(\xi_4 \nabla\xi_3 - \xi_3 \nabla\xi_4),
\end{aligned} \tag{A.2}$$

in which $L_i$ is the length of the $i$-th edge. The degrees of freedom of the 12 edge vector bases shown in (A.1) are located respectively at the following points in each element, with their corresponding projection directions $\hat{e}_i (i = 1, 2, ..., 12)$ defined as:

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

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 8 such bases, which are

$$\begin{aligned}
&\mathbf{N}_{13} = 4.5\xi_2\mathbf{W}_6, \quad \mathbf{N}_{14} = 4.5\xi_3\mathbf{W}_5 \\
&\mathbf{N}_{15} = 4.5\xi_3\mathbf{W}_3, \quad \mathbf{N}_{16} = 4.5\xi_4\mathbf{W}_2 \\
&\mathbf{N}_{17} = 4.5\xi_4\mathbf{W}_1, \quad \mathbf{N}_{18} = 4.5\xi_1\mathbf{W}_5 \\
&\mathbf{N}_{19} = 4.5\xi_1\mathbf{W}_4, \quad \mathbf{N}_{20} = 4.5\xi_2\mathbf{W}_2.
\end{aligned} \tag{A.4}$$

The locations $\mathbf{r}_{ei}(i = 13, 14, ..., 20)$ and corresponding unit vectors $\hat{e}_i$ associated with the above 8 face vector bases are:

$$\begin{aligned}
&\hat{e}_{13} = \hat{e}_{11}, \quad \mathbf{r}_{13} = (\xi_2 = \xi_3 = \xi_4 = 1/3, \xi_1 = 0) \\
&\hat{e}_{14} = \hat{e}_{9}, \quad \mathbf{r}_{14} = (\xi_2 = \xi_3 = \xi_4 = 1/3, \xi_1 = 0) \\
&\hat{e}_{15} = \hat{e}_{5}, \quad \mathbf{r}_{15} = (\xi_1 = \xi_3 = \xi_4 = 1/3, \xi_2 = 0) \\
&\hat{e}_{16} = \hat{e}_{3}, \quad \mathbf{r}_{16} = (\xi_1 = \xi_3 = \xi_4 = 1/3, \xi_2 = 0) \\
&\hat{e}_{17} = \hat{e}_{1}, \quad \mathbf{r}_{17} = (\xi_1 = \xi_2 = \xi_4 = 1/3, \xi_3 = 0) \\
&\hat{e}_{18} = \hat{e}_{9}, \quad \mathbf{r}_{18} = (\xi_1 = \xi_2 = \xi_4 = 1/3, \xi_3 = 0) \\
&\hat{e}_{19} = \hat{e}_{7}, \quad \mathbf{r}_{19} = (\xi_1 = \xi_2 = \xi_3 = 1/3, \xi_4 = 0) \\
&\hat{e}_{20} = \hat{e}_{3}, \quad \mathbf{r}_{20} = (\xi_1 = \xi_2 = \xi_3 = 1/3, \xi_4 = 0).
\end{aligned} \tag{A.5}$$

# B. FIRST-ORDER CURL-CONFORMING VECTOR BASIS FUNCTIONS IN TRIANGULAR PRISM ELEMENT

In a triangular prism element, there are 36 first-order vector bases [70]. Their definitions are given below.

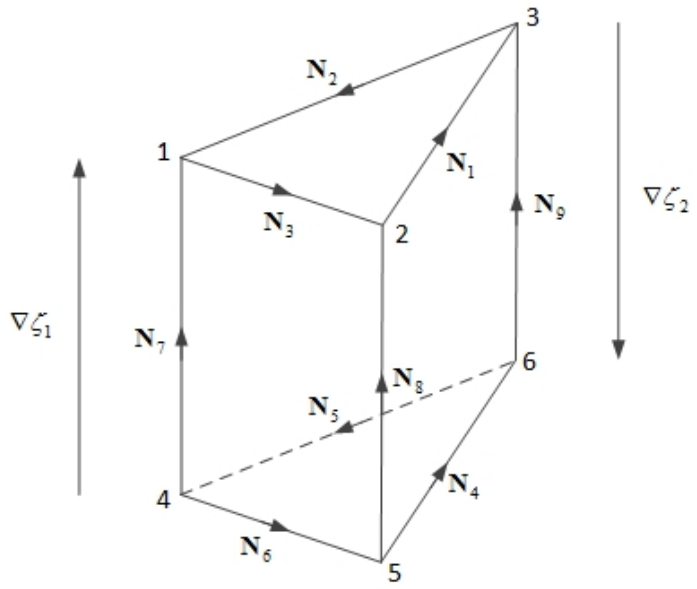



Fig. B.1. Illustration of the zeroth-order vector bases for triangular prism element

We first define the three zeroth-order vector basis functions for a 2-D triangular elements as follows

$$\begin{aligned} \mathbf{W}_1 &= l_1 \left(\xi_2 \nabla \xi_3 - \xi_3 \nabla \xi_2\right) \\ \mathbf{W}_2 &= l_2 \left(\xi_3 \nabla \xi_1 - \xi_1 \nabla \xi_3\right) \\ \mathbf{W}_3 &= l_3 \left(\xi_1 \nabla \xi_2 - \xi_2 \nabla \xi_1\right), \end{aligned} \tag{B.1}$$

where $l_i$ $(i = 1, 2, 3)$ is the length of $i$-th edge and $\xi_i$ $(i = 1, 2, 3)$ is the area coordinate.

Table B.1
Definition of the zeroth-order vector bases for triangular prism element

| Projection Direction | $\xi_1$ | $\xi_2$ | $\xi_3$ | $\zeta_1$ | $\zeta_2$ | Vector Basis |
|---|---|---|---|---|---|---|
| $\hat{e}_1 = \hat{t}_{23}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | 1 | 0 | $\mathbf{N}_1 = \zeta_1 \mathbf{W}_1$ |
| $\hat{e}_2 = \hat{t}_{31}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 1 | 0 | $\mathbf{N}_2 = \zeta_1 \mathbf{W}_2$ |
| $\hat{e}_3 = \hat{t}_{12}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 1 | 0 | $\mathbf{N}_3 = \zeta_1 \mathbf{W}_3$ |
| $\hat{e}_4 = \hat{t}_{56}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 1 | $\mathbf{N}_4 = \zeta_2 \mathbf{W}_1$ |
| $\hat{e}_5 = \hat{t}_{64}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | 0 | 1 | $\mathbf{N}_5 = \zeta_2 \mathbf{W}_2$ |
| $\hat{e}_6 = \hat{t}_{45}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | 0 | 1 | $\mathbf{N}_6 = \zeta_2 \mathbf{W}_3$ |
| $\hat{e}_7 = \hat{t}_{41}$ | 1 | 0 | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{N}_7 = h\xi_1 \nabla \zeta_1$ |
| $\hat{e}_8 = \hat{t}_{52}$ | 0 | 1 | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{N}_8 = h\xi_2 \nabla \zeta_1$ |
| $\hat{e}_9 = \hat{t}_{63}$ | 0 | 0 | 1 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{N}_9 = h\xi_3 \nabla \zeta_1$ |

Assume the height of the triangular prism element is $h$. $\zeta_1$ varies from 0 to 1, and is 1 on the face formed by node 1-3 while 0 on the other triangular face. Meanwhile, $\zeta_1 + \zeta_2 = 1$. With (B.1), we can summarize the definition of the 9 zeroth-order vector bases shown in Fig. B.1 with their corresponding location and projection directions $\hat{e}_i$ $(i = 1, 2, ..., 9)$ in Table B.1

Among the 36 first-order vector bases for triangular prism element, 18 of them are located on the edges. Their definitions $\mathbf{B}_i$ $(i = 1, 2, \ldots, 18)$ with corresponding locations and projection directions $\hat{u}_i$ $(i = 1, 2, \ldots, 18)$ are summarized in Table B.2.

Except for those vector bases on the edges, there also exist 4 vector bases on each side rectangular face, and their definitions are shown in Table B.3.

On each triangular face, there are also two vector bases located at the center of the face. Their definitions are given in Table B.4.

Finally, there are two vector bases located at the center of the triangular prism element. See Table B.5.

Table B.2
Definition of the 18 first-order vector bases located on the edges of triangular prism element

| Projection Direction | $\xi_1$ | $\xi_2$ | $\xi_3$ | $\zeta_1$ | $\zeta_2$ | Vector Basis |
|---|---|---|---|---|---|---|
| $\hat{u}_1 = \hat{t}_{23}$ | 0 | $\frac{2}{3}$ | $\frac{1}{3}$ | 1 | 0 | $\mathbf{B}_1 = (3\xi_2 - 1)(2\zeta_1 - 1)\mathbf{N}_1$ |
| $\hat{u}_2 = \hat{t}_{23}$ | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | 1 | 0 | $\mathbf{B}_2 = (3\xi_3 - 1)(2\zeta_1 - 1)\mathbf{N}_1$ |
| $\hat{u}_3 = \hat{t}_{31}$ | $\frac{1}{3}$ | 0 | $\frac{2}{3}$ | 1 | 0 | $\mathbf{B}_3 = (3\xi_3 - 1)(2\zeta_1 - 1)\mathbf{N}_2$ |
| $\hat{u}_4 = \hat{t}_{31}$ | $\frac{2}{3}$ | 0 | $\frac{1}{3}$ | 1 | 0 | $\mathbf{B}_4 = (3\xi_1 - 1)(2\zeta_1 - 1)\mathbf{N}_2$ |
| $\hat{u}_5 = \hat{t}_{12}$ | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 | 1 | 0 | $\mathbf{B}_5 = (3\xi_1 - 1)(2\zeta_1 - 1)\mathbf{N}_3$ |
| $\hat{u}_6 = \hat{t}_{12}$ | $\frac{1}{3}$ | $\frac{2}{3}$ | 0 | 1 | 0 | $\mathbf{B}_6 = (3\xi_2 - 1)(2\zeta_1 - 1)\mathbf{N}_3$ |
| $\hat{u}_7 = \hat{t}_{56}$ | 0 | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 | 1 | $\mathbf{B}_7 = (3\xi_2 - 1)(2\zeta_2 - 1)\mathbf{N}_4$ |
| $\hat{u}_8 = \hat{t}_{56}$ | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | 0 | 1 | $\mathbf{B}_8 = (3\xi_3 - 1)(2\zeta_2 - 1)\mathbf{N}_4$ |
| $\hat{u}_9 = \hat{t}_{64}$ | $\frac{1}{3}$ | 0 | $\frac{2}{3}$ | 0 | 1 | $\mathbf{B}_9 = (3\xi_3 - 1)(2\zeta_2 - 1)\mathbf{N}_5$ |
| $\hat{u}_{10} = \hat{t}_{64}$ | $\frac{2}{3}$ | 0 | $\frac{1}{3}$ | 0 | 1 | $\mathbf{B}_{10} = (3\xi_1 - 1)(2\zeta_2 - 1)\mathbf{N}_5$ |
| $\hat{u}_{11} = \hat{t}_{45}$ | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 | 0 | 1 | $\mathbf{B}_{11} = (3\xi_1 - 1)(2\zeta_2 - 1)\mathbf{N}_6$ |
| $\hat{u}_{12} = \hat{t}_{45}$ | $\frac{1}{3}$ | $\frac{2}{3}$ | 0 | 0 | 1 | $\mathbf{B}_{12} = (3\xi_2 - 1)(2\zeta_2 - 1)\mathbf{N}_6$ |
| $\hat{u}_{13} = \hat{t}_{41}$ | 1 | 0 | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\mathbf{B}_{13} = (2\xi_1 - 1)(3\zeta_2 - 1)\mathbf{N}_7$ |
| $\hat{u}_{14} = \hat{t}_{41}$ | 1 | 0 | 0 | $\frac{2}{3}$ | $\frac{1}{3}$ | $\mathbf{B}_{14} = (2\xi_1 - 1)(3\zeta_1 - 1)\mathbf{N}_7$ |
| $\hat{u}_{15} = \hat{t}_{52}$ | 0 | 1 | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\mathbf{B}_{15} = (2\xi_2 - 1)(3\zeta_2 - 1)\mathbf{N}_8$ |
| $\hat{u}_{16} = \hat{t}_{52}$ | 0 | 1 | 0 | $\frac{2}{3}$ | $\frac{1}{3}$ | $\mathbf{B}_{16} = (2\xi_2 - 1)(3\zeta_1 - 1)\mathbf{N}_8$ |
| $\hat{u}_{17} = \hat{t}_{63}$ | 0 | 0 | 1 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\mathbf{B}_{17} = (2\xi_3 - 1)(3\zeta_2 - 1)\mathbf{N}_9$ |
| $\hat{u}_{18} = \hat{t}_{63}$ | 0 | 0 | 1 | $\frac{2}{3}$ | $\frac{1}{3}$ | $\mathbf{B}_{18} = (2\xi_3 - 1)(3\zeta_1 - 1)\mathbf{N}_9$ |

Table B.3
Definition of the 12 first-order vector bases located on the side rectangular faces of triangular prism element

| Projection Direction | $\xi_1$ | $\xi_2$ | $\xi_3$ | $\zeta_1$ | $\zeta_2$ | Vector Basis |
|---|---|---|---|---|---|---|
| $\hat{u}_{19} = \hat{t}_{23}$ | 0 | $\frac{2}{3}$ | $\frac{1}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{19} = 4\zeta_2(3\xi_2 - 1)\mathbf{N}_1$ |
| $\hat{u}_{20} = \hat{t}_{23}$ | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{20} = 4\zeta_2(3\xi_3 - 1)\mathbf{N}_1$ |
| $\hat{u}_{21} = \hat{t}_{52}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{1}{3}$ | $\frac{2}{3}$ | $\mathbf{B}_{21} = 4\xi_3(3\zeta_2 - 1)\mathbf{N}_8$ |
| $\hat{u}_{22} = \hat{t}_{52}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\frac{2}{3}$ | $\frac{1}{3}$ | $\mathbf{B}_{22} = 4\xi_3(3\zeta_1 - 1)\mathbf{N}_8$ |
| $\hat{u}_{23} = \hat{t}_{31}$ | $\frac{1}{3}$ | 0 | $\frac{2}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{23} = 4\zeta_2(3\xi_3 - 1)\mathbf{N}_2$ |
| $\hat{u}_{24} = \hat{t}_{31}$ | $\frac{2}{3}$ | 0 | $\frac{1}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{24} = 4\zeta_2(3\xi_1 - 1)\mathbf{N}_2$ |
| $\hat{u}_{25} = \hat{t}_{63}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | $\frac{1}{3}$ | $\frac{2}{3}$ | $\mathbf{B}_{25} = 4\xi_1(3\zeta_2 - 1)\mathbf{N}_9$ |
| $\hat{u}_{26} = \hat{t}_{63}$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | $\frac{2}{3}$ | $\frac{1}{3}$ | $\mathbf{B}_{26} = 4\xi_1(3\zeta_1 - 1)\mathbf{N}_9$ |
| $\hat{u}_{27} = \hat{t}_{12}$ | $\frac{2}{3}$ | $\frac{1}{3}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{27} = 4\zeta_2(3\xi_1 - 1)\mathbf{N}_3$ |
| $\hat{u}_{28} = \hat{t}_{12}$ | $\frac{1}{3}$ | $\frac{2}{3}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{28} = 4\zeta_2(3\xi_2 - 1)\mathbf{N}_3$ |
| $\hat{u}_{29} = \hat{t}_{41}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\mathbf{B}_{29} = 4\xi_2(3\zeta_2 - 1)\mathbf{N}_7$ |
| $\hat{u}_{30} = \hat{t}_{41}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 0 | $\frac{2}{3}$ | $\frac{1}{3}$ | $\mathbf{B}_{30} = 4\xi_2(3\zeta_1 - 1)\mathbf{N}_7$ |

Table B.4
Definition of the 4 first-order vector bases located on the triangular faces of triangular prism element

| Projection Direction | $\xi_1$ | $\xi_2$ | $\xi_3$ | $\zeta_1$ | $\zeta_2$ | Vector Basis |
|---|---|---|---|---|---|---|
| $\hat{u}_{31} = \hat{t}_{56}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | 0 | 1 | $\mathbf{B}_{31} = 4.5\xi_1(2\zeta_2 - 1)\mathbf{N}_4$ |
| $\hat{u}_{32} = \hat{t}_{64}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | 0 | 1 | $\mathbf{B}_{32} = 4.5\xi_2(2\zeta_2 - 1)\mathbf{N}_5$ |
| $\hat{u}_{33} = \hat{t}_{23}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | 1 | 0 | $\mathbf{B}_{33} = 4.5\xi_1(2\zeta_1 - 1)\mathbf{N}_1$ |
| $\hat{u}_{34} = \hat{t}_{31}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | 1 | 0 | $\mathbf{B}_{34} = 4.5\xi_2(2\zeta_1 - 1)\mathbf{N}_2$ |

Table B.5
Definition of the 2 first-order vector bases located at the center of triangular prism element

| Projection Direction | $\xi_1$ | $\xi_2$ | $\xi_3$ | $\zeta_1$ | $\zeta_2$ | Vector Basis |
|---|---|---|---|---|---|---|
| $\hat{u}_{35} = \hat{t}_{23}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{35} = 18\xi_1\zeta_2\mathbf{N}_1$ |
| $\hat{u}_{36} = \hat{t}_{31}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{2}$ | $\frac{1}{2}$ | $\mathbf{B}_{36} = 18\xi_2\zeta_2\mathbf{N}_2$ |

VITA

# VITA

Jin Yan received the B.S. degree in Electronic Information Engineering from the University of Science and Technology of China, Hefei, China, in 2012. Since then, she has been working towards the Ph.D. degree in Electrical Engineering with Professor Dan Jiao in the On-Chip Electromagnetic Lab, Purdue University, West Lafayette, IN, USA. Her research interests include computational electromagnetics, high-performance VLSI CAD, and fast and high-capacity numerical methods. She was the recipient of an Honorable Mention Award of the IEEE International Symposium on Antennas and Propagation in 2015 and a Best Student Paper Award Finalist from the IEEE MTT-S International Microwave Symposium in 2016.