Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **8**, No. 3, pp. 353-385

DOI: 10.4208/aamm.2014.m811
June 2016

Time-Domain Numerical Solutions of Maxwell Interface Problems with Discontinuous Electromagnetic Waves

Ya Zhang[1], Duc Duy Nguyen[2], Kewei Du[1], Jin Xu[1] and Shan Zhao[2,*]

[1] *Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*
[2] *Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA*

Received 3 November 2014; Accepted (in revised version) 6 February 2015

Abstract. This paper is devoted to time domain numerical solutions of two-dimensional (2D) material interface problems governed by the transverse magnetic (TM) and transverse electric (TE) Maxwell's equations with discontinuous electromagnetic solutions. Due to the discontinuity in wave solutions across the interface, the usual numerical methods will converge slowly or even fail to converge. This calls for the development of advanced interface treatments for popular Maxwell solvers. We will investigate such interface treatments by considering two typical Maxwell solvers – one based on collocation formulation and another based on Galerkin formulation. To restore the accuracy reduction of the collocation finite-difference time-domain (FDTD) algorithm near an interface, the physical jump conditions relating discontinuous wave solutions on both sides of the interface must be rigorously enforced. For this purpose, a novel matched interface and boundary (MIB) scheme is proposed in this work, in which new jump conditions are derived so that the discontinuous and staggered features of electric and magnetic field components can be accommodated. The resulting MIB time-domain (MIBTD) scheme satisfies the jump conditions locally and suppresses the staircase approximation errors completely over the Yee lattices. In the discontinuous Galerkin time-domain (DGTD) algorithm – a popular Galerkin Maxwell solver, a proper numerical flux can be designed to accurately capture the jumps in the electromagnetic waves across the interface and automatically preserves the discontinuity in the explicit time integration. The DGTD solution to Maxwell interface problems is explored in this work, by considering a nodal based high order discontinuous Galerkin method. In benchmark TM and TE tests with analytical solutions, both MIBTD and DGTD schemes achieve the second order of accuracy in solving circular interfaces. In comparison, the numerical convergence of the MIBTD method is slightly more uniform, while the DGTD method is more flexible and robust.

AMS subject classifications: 65M06, 65M60, 78M10, 78M20

Key words: Maxwell's equations, finite-difference time-domain (FDTD), discontinuous Galerkin time-domain (DGTD), transverse magnetic (TM) and transverse electric (TE) systems, high order interface treatments, matched interface and boundary (MIB).

*Corresponding author.
Email: szhao@ua.edu (S. Zhao)

 ©2016 Global Science Press

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

1 Introduction

When the permittivity and permeability coefficients are discontinuous across a material interface separating two dielectric media, the electric and magnetic field components could be discontinuous. Without a proper interface treatment, the time domain numerical solution of Maxwell's equations that govern the propagation and scattering of electromagnetic waves in nonhomogeneous media converges slowly or even fails to converge [12]. Because the material interfaces are omnipresent in optical devices, microwave circuits, antennas, aircraft radar signature, nano/micro electric devices and telecommunication chips, the development of innovative computational methods for dealing with electromagnetic interface problems with discontinuous solutions has received much attention in recent years. Various different interface strategies have been developed in the computational electromagnetics (CEM) to cope with the problems caused by the loss of solution regularity and the complex geometry of the material interface [12].

It is well known that the finite-difference time-domain (FDTD) method based on the Yee lattice has been a main workhorse of the CEM in the time domain [34], owing to its simplicity, free of dissipative error and having very low cost per grid node. However, the standard FDTD algorithm suffers from a large accuracy reduction at a dielectric interface, due to not only a staircase representation for complex geometry, but also the lack of the enforcement of solution jumps in the numerical discretization. To overcome this difficulty, several embedded FDTD methods [3, 8, 12, 36, 37, 40, 41, 46] have been successfully developed to restore the accuracy near a dielectric interface. Computationally, some sophisticated interface treatments are conducted near the interface or on a one-dimensional smaller set of nodes to rigorously impose the jump conditions in the FDTD discretizations, while away from the interface, the standard FDTD scheme can be employed. While maintaining the simplicity and computational efficiency of the Yee scheme [34], the embedded FDTD methods can fully restore second order accuracy, even in case of curved boundaries and interfaces, by using a simple Cartesian grid. Nevertheless, Maxwell interface problems with continuous wave solutions are normally considered in these embedded FDTD methods. The discontinuous Maxwell interface problem to be attacked in this work is less well studied in the FDTD literature.

Recently, we have developed a family of finite difference algorithms based on the matched interface and boundary (MIB) method [42, 47] for solving regular dielectric interface problems [42, 43] and dispersive interface problems [29, 30, 44]. The MIB interface treatments are quite flexible, through the introduction of fictitious values and an iterative use of zeroth and first order jump conditions [42, 47]. The MIB modeling is systematically carried out and can be made to arbitrarily high order in principle in the presence of straight material interfaces. Orders up to 12 have been achieved numerically for solving Maxwell's equations [42]. When dispersive materials are considered, the material constitutive coefficients are functions of frequency so that the jumps in electromagnetic fields become time variant across dispersive interfaces. The MIB time-domain (MIBTD) method [29, 30, 44] is the only known second order accurate FDTD algorithm for solving

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

dispersive interface problems, thanks to its capability in modeling time dependent jump conditions. However, in all previous MIBTD algorithms [29, 42, 44], only simple interface problems with continuous electric and magnetic field components are considered. The generalization of the existing MIB scheme to treat discontinuous Maxwell interface problem remains to be unsolved.

Using unstructured grids, the finite element methods (FEMs) are some of the most flexible methods for solving Maxwell interface problems. The edge element method developed in 1980s [2, 28, 31, 32] is the most well known FEM in the CEM literature, which assigns degrees of freedom to the edges of the vector elements so that the edge element method is free of spurious solutions and is very flexible in imposing boundary conditions at material interfaces. Recently, a variety of finite element time domain (FETD) methods [9, 18, 33] have been developed for solving Maxwell's equations in the time domain. Nevertheless, the classical FETD methods may lead to a high computational cost since a matrix inversion is required at each time step. The modern approach to bypass this matrix inversion is using the discontinuous Galerkin time-domain (DGTD) methods [4–7, 10, 11, 13, 21, 23, 26, 27]. Based on discontinuous basis functions, the DGTD methods are locally conservative, stable and high-order accurate. In the discontinuous Galerkin variational formulation, it is quite easy to handle complex geometries, irregular meshes with hanging nodes and approximations that have polynomials of different degrees in different elements. These nice properties attribute to the popularity of the DGTD methods for various applications in the CEM. In particular, the DGTD methods have achieved a great success in modeling complex materials, such as dispersive media [14, 15, 17, 19, 24, 25, 35] and metamaterials [20, 22]. It will be of great interests to investigate the performance of the DGTD methods for solving discontinuous Maxwell interface problems, because the DGTD methods are by design using discontinuous approximation functions in different elements. By properly choosing the numerical flux, the jumps in the electromagnetic waves across the interface can be accurately captured and well preserved in the DGTD computations.

The goal of this paper is to carry out a comparative study of MIBTD and DGTD methods for solving Maxwell interface problems, with a particular stress on problems in which the magnetic and/or electric field components are discontinuous. We will focus on two types of two-dimensional (2D) systems, i.e., the transverse magnetic (TM) and transverse electric (TE) Maxwell's equations. A circular interface will be considered in both systems with discontinuous constitutive coefficients. Analytical solutions are available so that the spatial order of accuracy can be numerically evaluated.

For the MIBTD method, a novel formulation of jump conditions will be proposed, so that the discontinuous and staggered features of electric and magnetic field components can be accommodated. This facilitates the implementation of the jump conditions in the FDTD discretization and suppresses the staircase approximation errors completely over the Yee lattices. The present study represents a considerable progress on the development of MIBTD methods for solving Maxwell's equations, because the proposed MIBTD method is the first of its kind that can handle discontinuous electromagnetic waves.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

The DGTD method being tested in this work is based on the nodal based discontinuous Galerkin approach, originally proposed by Hesthaven and Warburton [13]. We have developed our DGTD package fully in C++ environment. It has also been parallelized to make use of a large number of processors for large-scale simulations. The public software CUBIT has been used to generate the mesh inside and outside of interface for different materials. This facilitates to define different material properties in different regions. Several different methods to partition the mesh have been developed, so that each processor will handle nearly the same number of elements and minimal communications can be achieved to reach good scalability. The present study provides excellent benchmark tests to validate this new package and more sophisticated algorithms can be easily added on it.

This paper is organized as follows. The Maxwell's equations and the interface setting will be given in Section 2. The proposed MIBTD algorithm will be presented in Section 3. Section 4 devotes to an introduction of the nodal based DGTD method. The simulation results of both MIBTD and DGTD approaches will be reported in Section 5. The stability, accuracy and efficiency of the algorithms will be examined using several benchmark examples. Finally, this paper ends with a conclusion.

2 Maxwell interface problem

In three dimensions (3D), Maxwell's equations that govern all macroscopic electromagnetic phenomena can be given as

$$\frac{\partial \vec{D}}{\partial t} = \nabla \times \vec{H} - \vec{J}, \qquad \nabla \cdot \vec{D} = \rho, \tag{2.1a}$$

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \vec{E}, \qquad \nabla \cdot \vec{B} = 0, \tag{2.1b}$$

where $\vec{D}(\vec{x},t)$ and $\vec{B}(\vec{x},t)$ are, respectively, the electric and magnetic flux densities, while $\vec{E}(\vec{x},t)$ and $\vec{H}(\vec{x},t)$ are, respectively, the electric and magnetic field intensities. For simplicity, we consider the case with the electric charge density ρ and the electric current density $\vec{J}$ being zero. Consider linear and isotropic materials with the constitutive relations

$$\vec{D} = \epsilon \vec{E}, \quad \vec{B} = \mu \vec{H}, \tag{2.2}$$

where ϵ and μ are, respectively, the electric permittivity and magnetic permeability. Here, a nondimensional form of the equations is considered, i.e., $\epsilon = \mu = 1$ in free space.

In this paper, we will focus on two-dimensional (2D) Maxwell's systems by assuming that the structure being considered is invariant in the z-direction and the incident wave is also uniform in the z-direction. With this assumption, all partial derivatives of the fields of 3D Maxwell's equations with respect to z are zero. The resulting 2D equations can be grouped into two independent sets, i.e., transverse magnetic (TM) mode and transverse

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

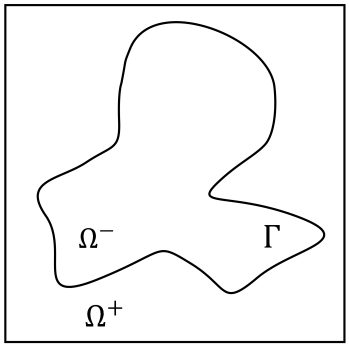


Figure 1: A curved interface Γ separating two media over the domain Ω.

electric (TE) mode as follows

$$\text{TM}_z \text{ mode:} \quad \frac{\partial E_z}{\partial t}=\frac{1}{\epsilon}\Big(\frac{\partial H_y}{\partial x}-\frac{\partial H_x}{\partial y}\Big), \quad \frac{\partial H_y}{\partial t}=\frac{1}{\mu}\frac{\partial E_z}{\partial x}, \quad \frac{\partial H_x}{\partial t}=-\frac{1}{\mu}\frac{\partial E_z}{\partial y}, \tag{2.3a}$$

$$\text{TE}_z \text{ mode:} \quad \frac{\partial H_z}{\partial t}=\frac{1}{\mu}\Big(\frac{\partial E_x}{\partial y}-\frac{\partial E_y}{\partial x}\Big), \quad \frac{\partial E_y}{\partial t}=-\frac{1}{\epsilon}\frac{\partial H_z}{\partial x}, \quad \frac{\partial E_x}{\partial t}=\frac{1}{\epsilon}\frac{\partial H_z}{\partial y}, \tag{2.3b}$$

where $[E_x,E_y,E_z]^T=\vec{E}$ and $[H_x,H_y,H_z]^T=\vec{H}$ are, respectively, the electric and magnetic fields components in the corresponding Cartesian directions.

The material interface is omnipresent in electromagnetics, such as coatings used on the surface of an aircraft for minimizing the radar cross section (RCS) and gaps in dielectric materials. In the current 2D setting, we will consider a general Maxwell interface problem as illustrated in Fig. 1. For simplicity, the domain Ω is assumed to be a rectangle one $\Omega=[a,b]\times[c,d]$, with proper boundary conditions prescribed for u on $\partial\Omega$. Across a material interface Γ separating two media Ω^- and Ω^+, material coefficients ϵ and μ could be discontinuous. We define the jump of a function $u(x,y)$ at Γ to be $[u]:=u^+-u^-$. The superscript, $-$ or $+$, denotes the limiting value of function from inside or outside of the interface. Without the loss of generality, the subdomain Ω^+ is assumed to be the air or vacuum with $\epsilon^+=\mu^+=1$, whereas the inside subdomain Ω^- is assigned to be a dielectric medium.

Across the material interface, the field solutions on both media are physically related through the jump conditions

$$\vec{n}\times(\vec{E}^+-\vec{E}^-)=0, \qquad \vec{n}\cdot(\epsilon^+\vec{E}^+-\epsilon^-\vec{E}^-)=0, \tag{2.4a}$$

$$\vec{n}\times(\vec{H}^+-\vec{H}^-)=0, \qquad \vec{n}\cdot(\mu^+\vec{H}^+-\mu^-\vec{H}^-)=0, \tag{2.4b}$$

where $\vec{n}$ is the unit normal vector to the interface, pointing from Ω^- to Ω^+. These zeroth order jump conditions suggest that the tangential components of $\vec{E}$ and $\vec{H}$ are continuous, while their normal components could be discontinuous. Combining the zeroth order

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

jump conditions (2.4) with Maxwell's equations (2.3a) and (2.3b), we can derive first order jump conditions describing the regularities in the first derivatives of $\vec{E}$ and $\vec{H}$ [42,43]. Higher order jump conditions can be derived in a similar manner, but become very complicated so that they are usually skipped in the numerical interface treatments [42].

3 Matched interface and boundary (MIB) algorithm

In this section, we propose a novel MIB time-domain (MIBTD) algorithm for solving the discontinuous Maxwell interface problem. Previously, we have developed several MIBTD schemes for solving regular dielectric interface problems [42,43] and dispersive interface problems [29,44]. However, the electromagnetic fields are assumed to be continuous across the interface in these existing studies. We note that for the scalar elliptic and parabolic interface problems, robust MIB schemes [45,47] have been constructed to handle discontinuous solutions. Nevertheless, these discontinuous interface treatments can not be directly applied to the present Maxwell interface problem, which is of a vectorized nature, i.e., different field components will be coupled through the jump conditions. Moreover, in the Yee mesh, the field components are located in staggered positions. This introduces additional difficulties for solving Maxwell interface problems.

3.1 Finite difference time domain (FDTD) discretization based on MIB

Without the loss of generality, we will present the new MIBTD algorithm by considering the TM system (2.3a). The MIBTD method for solving the TE system (2.3b) can be similarly constructed.

Denote the time increment to be Δt. Both electric and magnetic components will be assumed at the same time instant $t_k = k\Delta t$ in the MIBTD algorithm. The classical fourth order Runge-Kutta scheme will be employed to integrate the TM system (2.3a) from t_k to t_{k+1}. A uniform staggered grid shown in Fig. 2 is used in the MIBTD algorithm. Denote Δx and Δy to be the spacing in x and y direction, respectively. To facilitate the following discussions, we adopt the following notations: $E^z_{i,j} := E_z(x_i, y_j, t_k)$, $H^x_{i,j+\frac{1}{2}} := H_x(x_i, y_{j+\frac{1}{2}}, t_k)$

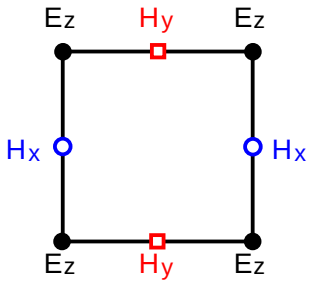


Figure 2: The staggered grid system.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

and $H^y_{i+\frac{1}{2},j} := H_y(x_{i+\frac{1}{2}},y_j,t_k)$. For nodes away from the interface, the central finite difference approximations as in the standard FDTD algorithm are employed for the spatial discretization, e.g.,

$$\frac{\partial H_y}{\partial x}\Big|_{(x_i,y_j)} \approx \frac{H^y_{i+\frac{1}{2},j} - H^y_{i-\frac{1}{2},j}}{\Delta x}. \tag{3.1}$$

For nodes near Γ, the MIB interface treatment [42, 47] will be carried out to correct finite difference approximations via rigorously imposing the jump conditions. A universal rule here is that to approximate function or its derivatives on one side of interface, one never *directly* refers to function values from the other side. Instead, fictitious values form the other side of the interface will be supplied. For example, suppose that Γ cuts the grid line $y=y_j$ in between x_i and $x_{i+\frac{1}{2}}$. We need to determine a fictitious value of H_y at $(x_{i+\frac{1}{2}},y_j)$. We denote it as $f^y_{i+\frac{1}{2},j}$. Then the approximation (3.1) shall be modified to be

$$\frac{\partial H_y}{\partial x}\Big|_{(x_i,y_j)} \approx \frac{f^y_{i+\frac{1}{2},j} - H^y_{i-\frac{1}{2},j}}{\Delta x}. \tag{3.2}$$

Likewise, if Γ passes the line $y=y_j$ in between $x_{i-\frac{1}{2}}$ and x_i, (3.1) will be changed to

$$\frac{\partial H_y}{\partial x}\Big|_{(x_i,y_j)} \approx \frac{H^y_{i+\frac{1}{2},j} - f^y_{i-\frac{1}{2},j}}{\Delta x}. \tag{3.3}$$

These corrected approximations maintain the second order of accuracy, provided that fictitious values are accurately estimated by using the jump conditions (2.4).

3.2 Interface jump conditions in a local coordinate

Since jump conditions (2.4) are physically defined in terms of normal and tangential directions, we first derive jump conditions in a local coordinate. Consider a point $P(x_o,y_o)$ located on the interface Γ. We define a local coordinate system $(\vec{n},\vec{\tau})$ at P, see Fig. 3. Any nearby point (x,y) will have local coordinate values

$$n = \cos\theta x + \sin\theta y, \quad \tau = -\sin\theta x + \cos\theta y, \tag{3.4}$$

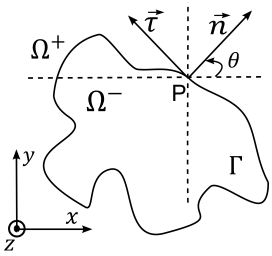


Figure 3: Local normal and tangential coordinate of a point on the interface Γ.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

where $0\leq\theta<2\pi$ is the angle between the positive x-direction and the normal vector $\vec{n}$ at P. It it also worthwhile mentioning the derivative operators on this local coordinate

$$\frac{\partial}{\partial n}=\cos\theta\frac{\partial}{\partial x}+\sin\theta\frac{\partial}{\partial y},\quad \frac{\partial}{\partial\tau}=-\sin\theta\frac{\partial}{\partial x}+\cos\theta\frac{\partial}{\partial y}. \tag{3.5}$$

On the local grid system, jump conditions (2.4) give us the following zeroth order jump conditions

$$[E_z]=0,\quad [E_\tau]=0,\quad [\epsilon E_n]=0,\quad [H_z]=0,\quad [H_\tau]=0,\quad [\mu H_n]=0. \tag{3.6}$$

First order jump conditions of all fields can be implied based on the zeroth order conditions (3.6) and Maxwell's equations [29,30,42]. For the E_z component, one jump condition can be derived by taking the derivative with respect to the tangent direction of the first equation in (3.6)

$$\left[\frac{\partial E_z}{\partial\tau}\right]=0. \tag{3.7}$$

To seek for another jump condition of this field, we rewrite the second equation of (2.3a) in local coordinate and then apply the jump operators on it to obtain

$$\left[\frac{1}{\mu}\frac{\partial E_z}{\partial n}\right]=\left[\frac{\partial H_\tau}{\partial t}\right]=0, \tag{3.8}$$

where we make use of the fact $[H_\tau]=0$. Concerning the jump conditions of the magnetic component, one takes the derivative with respect to τ of the two last conditions in (3.6) to achieve

$$\left[\frac{\partial H_\tau}{\partial\tau}\right]=0,\quad \left[\frac{\partial(\mu H_n)}{\partial\tau}\right]=0. \tag{3.9}$$

Another jump condition of magnetic field can be derived by employing the divergence free law

$$\nabla\cdot(\mu H)=\frac{\partial(\mu H_n)}{\partial n}+\frac{\partial(\mu H_\tau)}{\partial\tau}+\frac{\partial(\mu H_z)}{\partial z}=0. \tag{3.10}$$

In the TM mode, all of the fields are uniform along the z-direction. Therefore, their derivatives with respect to z can be omitted. Consequently, Eq. (3.10) becomes in a simpler form

$$\frac{\partial(\mu H_n)}{\partial n}+\frac{\partial(\mu H_\tau)}{\partial\tau}=0. \tag{3.11}$$

We take jump values on both hand sides of Eq. (3.11)

$$\left[\frac{\partial(\mu H_\tau)}{\partial\tau}\right]+\left[\frac{\partial(\mu H_n)}{\partial n}\right]=0. \tag{3.12}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

One multiplies the first equation of (2.3a) by μ and then takes jump operators on it to give rise to the last jump condition of H field

$$\left[\frac{\partial(\mu H_\tau)}{\partial n}\right]-\left[\frac{\partial(\mu H_n)}{\partial \tau}\right]=\left[\frac{\partial(\mu\epsilon E_z)}{\partial t}\right]. \tag{3.13}$$

We note that the condition (3.13) is a new condition which has never been used in the previous MIBTD schemes. In its present form, (3.13) contains a time dependent part. Nevertheless, we will show later that this term can be estimated based on known H_x and H_y values at time t_k.

3.3 Interface jump conditions in the Cartesian coordinate

Since all the FDTD spatial discretization is conducted in the Cartesian directions, we need to transform the jump conditions in the local coordinate into the Cartesian ones. Moreover, a key idea in the MIB interface treatment is to decompose the 2D jump conditions so that they can be imposed in a 1D manner [43,47]. We will consider such decompositions, while taking care of the coordinate transformations.

We first consider the zeroth order jump conditions for Cartesian field components. For E_z, we simply have

$$E_z^+-E_z^-=0. \tag{3.14}$$

In order to derive the jump conditions for H_x and H_y, one can use coordinate transformation mentioned in (3.4) to translate the zeroth order jump conditions $[H_\tau]$=0 and $[\mu H_n]$=0 into

$$0=[H_\tau]=-\sin\theta H_x^+ +\cos\theta H_y^+ +\sin\theta H_x^- -\cos\theta H_y^-, \tag{3.15a}$$

$$0=[\mu H_n]=\cos\theta\mu^+ H_x^+ +\sin\theta\mu^+ H_y^+ -\cos\theta\mu^- H_x^- -\sin\theta\mu^- H_y^-. \tag{3.15b}$$

Because our incident waves propagate from the positive side, H_x^- will be eliminated from (3.15*a*) and (3.15*b*) to obtain

$$C_x^+ H_x^+ +C_y^+ H_y^+ -C_x^- H_y^-=0, \tag{3.16}$$

in which

$$C_x^+=(\mu^+-\mu^-)\cos\theta\sin\theta,\quad C_y^+=\sin^2\theta\mu^+ +\cos^2\theta\mu^- \quad\text{and}\quad C_y^-=\mu^-.$$

Similarly, the new jump condition will be derived by canceling H_y^- from (3.15*a*) and (3.15*b*)

$$D_x^+ H_x^+ +D_y^+ H_y^+ -D_x^- H_x^-=0, \tag{3.17}$$

where

$$D_x^+=\sin^2\theta\mu^- +\cos^2\theta\mu^+,\quad D_y^+=(\mu^+-\mu^-)\sin\theta\cos\theta \quad\text{and}\quad D_x^-=\mu^-.$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Following the implementation given in the Appendix A, we can attain the following first order jump conditions for E_z

$$A_x^+\left(\frac{\partial E_z}{\partial x}\right)^+ + A_y^+\left(\frac{\partial E_z}{\partial y}\right)^+ - A_x^-\left(\frac{\partial E_z}{\partial x}\right)^- = 0, \tag{3.18a}$$

$$B_x^+\left(\frac{\partial E_z}{\partial x}\right)^+ + B_y^+\left(\frac{\partial E_z}{\partial y}\right)^+ - B_y^-\left(\frac{\partial E_z}{\partial y}\right)^- = 0, \tag{3.18b}$$

with the coefficients given in Appendix A. We note that the zeroth jump condition (3.14) and two first order jump conditions (3.18a) and (3.18b) for E_z are in the standard form, as discussed in the classical MIB scheme [47]. Moreover, these conditions only involves E_z field over integer grid nodes (x_i, y_j) on the Yee lattice (see Fig. 2). Therefore, the MIB modeling of E_z can be simply handled by the classical MIB scheme [47].

The situation for H_x and H_y becomes much more complicated. To attain the first order jump conditions for H_x and H_y, we again employ the coordinate transformations defined in (3.4) and (3.5). For the condition (3.13), we have

$$\begin{aligned}\left[\frac{\partial(\mu H_y)}{\partial x}\right] - \left[\frac{\partial(\mu H_x)}{\partial y}\right] &= \left[\frac{\partial(\mu\epsilon E_z)}{\partial t}\right] = (\mu^+\epsilon^+ - \mu^-\epsilon^-)\frac{\partial E_z^+}{\partial t} \\ &= \frac{\mu^+\epsilon^+ - \mu^-\epsilon^-}{\epsilon^+}\left\{\left(\frac{\partial H_y}{\partial x}\right)^+ - \left(\frac{\partial H_x}{\partial y}\right)^+\right\}.\end{aligned} \tag{3.19}$$

Here we have utilized the equation $[E_z] = 0$ and the TM Maxwell's equations to simplify the right hand side. The non-homogeneous term indeed becomes time independent, as we mentioned above. As discussed in Appendix B, we finally end up with four first order jump conditions for magnetic components

$$C_{xy}^+\left(\frac{\partial H_x}{\partial y}\right)^+ + C_{yx}^+\left(\frac{\partial H_y}{\partial x}\right)^+ + C_{yy}^+\left(\frac{\partial H_y}{\partial y}\right)^+ = C_{yx}^-\left(\frac{\partial H_y}{\partial x}\right)^-, \tag{3.20a}$$

$$D_{xx}^+\left(\frac{\partial H_x}{\partial x}\right)^+ + D_{xy}^+\left(\frac{\partial H_x}{\partial y}\right)^+ + D_{yx}^+\left(\frac{\partial H_y}{\partial x}\right)^+ = D_{xx}^-\left(\frac{\partial H_x}{\partial x}\right)^-, \tag{3.20b}$$

$$\hat{C}_{xy}^+\left(\frac{\partial H_x}{\partial y}\right)^+ + \hat{C}_{yx}^+\left(\frac{\partial H_y}{\partial x}\right)^+ + \hat{C}_{yy}^+\left(\frac{\partial H_y}{\partial y}\right)^+ = \hat{C}_{yy}^-\left(\frac{\partial H_y}{\partial y}\right)^-, \tag{3.20c}$$

$$\hat{D}_{xx}^+\left(\frac{\partial H_x}{\partial x}\right)^+ + \hat{D}_{xy}^+\left(\frac{\partial H_x}{\partial y}\right)^+ + \hat{D}_{yx}^+\left(\frac{\partial H_y}{\partial x}\right)^+ = \hat{D}_{xy}^-\left(\frac{\partial H_x}{\partial y}\right)^-, \tag{3.20d}$$

with the coefficients given in Appendix B.

We note that jump conditions for H_x and H_y are mixed together in these equations. This is a feature not seeing in scalar elliptic and parabolic interface problems [45, 47]. A novel MIB scheme has to be developed to address this issue. In the proposed MIB scheme for the TM equation, jump conditions (3.16), (3.17), (3.20a) and (3.20b) are used for correcting the derivatives of magnetic fields along the x direction, while jump conditions (3.16), (3.17), (3.20c) and (3.20d) are applied for seeking fictitious values of magnetic fields in the y direction. The jump conditions for the TE mode can be similarly derived.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

3.4 Jump conditions implementation

In this subsection, we discuss how to implement the jump conditions derived above in the MIB scheme to determine the necessary fictitious values for the FDTD discretization. To save the space, only MIB treatment for the TM mode is illustrated here. For the E_z component of the TM mode, the jump conditions are standard, so that the classical MIB scheme [47] can be simply applied. A new MIB scheme is proposed to treat H_x and H_y in this subsection.

The essence of the MIB scheme is to impose jump conditions in a 1D manner. Thus, the MIB treatments in x and y directions are essentially the same. It is sufficient to discuss the MIB treatment of H_x and H_y along the x direction in the present study. The modeling along the y direction can be similarly carried out. For the x direction, a group of jump conditions including Eqs. (3.16), (3.17), (3.20a) and (3.20b), will be studied. As in other Cartesian grid methods, in the MIB scheme, grid points are classified as irregular grid points if the standard finite approximation refers to a node from the other side of the interface. Otherwise they will be classified as regular grid points. For every irregular point, a fictitious value needs to be calculated through enforcing jump conditions.

A particular local grid topology is shown in Fig. 4. Here, we consider an interface point (x_o,y_o) on the interface Γ. In the vicinity of (x_o,y_o), the nodes $(x_{i-1/2},y_j)$ and $(x_{i+1/2},y_j)$ are the irregular grid points for H_y. We suppose that both $(x_{i-1/2},y_j)$ and (x_i,y_j) belong to Ω^- and $(x_{i+1/2},y_j)$ belongs to Ω^+. See Fig. 4. Following the notation defined above, we demonstrate how to determine four fictitious values $f^y_{i-1/2,j}$, $f^y_{i+1/2,j}$, $f^x_{i,j}$ and $f^x_{i+1,j}$ for the present situation.

For this purpose, we discretize four jump conditions (3.16), (3.17), (3.20a) and (3.20b) by following the universal rule of MIB scheme, i.e., never referring to function values on

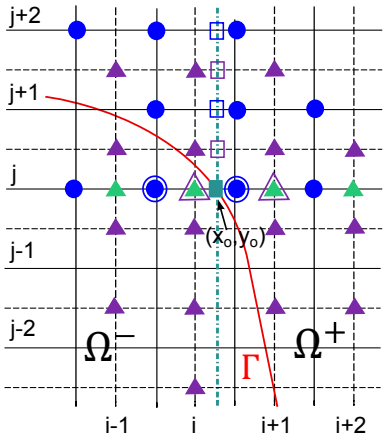


Figure 4: Illustration of the MIB grid partition. Filled circles: grid nodes for H_y; Fill triangles: nodes for H_x; Open circles: fictitious nodes for H_y; Open triangle: fictitious nodes for H_x; Open squares: auxiliary nodes for H_y and H_x; Square: interface node (x_o,y_o).

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

both hand sides for a single derivative approximation

$$C_x^+\left(\tilde{w}_{0,i}^+ f_{i,j}^x+\tilde{w}_{0,i+1}^+ H_{i+1,j}^x+\tilde{w}_{0,i+2}^+ H_{i+2,j}^x\right)+C_y^+\left(w_{0,i-\frac{1}{2}}^+ f_{i-\frac{1}{2},j}^y+w_{0,i+\frac{1}{2}}^+ H_{i+\frac{1}{2},j}^y+w_{0,i+\frac{3}{2}}^+ H_{i+\frac{3}{2},j}^y\right)$$
$$=C_y^-\left(w_{0,i-\frac{3}{2}}^- H_{i-\frac{3}{2},j}^y+w_{0,i-\frac{1}{2}}^- H_{i-\frac{1}{2},j}^y+w_{0,i+\frac{1}{2}}^- f_{i+\frac{1}{2},j}^y\right), \tag{3.21a}$$

$$D_x^+\left(\tilde{w}_{0,i}^+ f_{i,j}^x+\tilde{w}_{0,i+1}^+ H_{i+1,j}^x+\tilde{w}_{0,i+2}^+ H_{i+2,j}^x\right)+D_y^+\left(w_{0,i-\frac{1}{2}}^+ f_{i-\frac{1}{2},j}^y+w_{0,i+\frac{1}{2}}^+ H_{i+\frac{1}{2},j}^y+w_{0,i+\frac{3}{2}}^+ H_{i+\frac{3}{2},j}^y\right)$$
$$=D_x^-\left(\tilde{w}_{0,i-1}^- H_{i-1,j}^x+\tilde{w}_{0,i}^- H_{i,j}^x+\tilde{w}_{0,i+1}^- f_{i+1,j}^x\right), \tag{3.21b}$$

$$C_{xy}^+\left(\frac{\partial H_x}{\partial y}\right)^+\Big|_{x_o,y_o}+C_{yx}^+\left(w_{1,i-\frac{1}{2}}^+ f_{i-\frac{1}{2},j}^y+w_{1,i+\frac{1}{2}}^+ H_{i+\frac{1}{2},j}^y+w_{1,i+\frac{3}{2}}^+ H_{i+\frac{3}{2},j}^y\right)+C_{yy}^+\left(\frac{\partial H_y}{\partial y}\right)^+\Big|_{x_o,y_o}$$
$$=C_{yx}^-\left(w_{1,i-\frac{3}{2}}^- H_{i-\frac{3}{2},j}^y+w_{1,i-\frac{1}{2}}^- H_{i-\frac{1}{2},j}^y+w_{1,i+\frac{1}{2}}^- f_{i+\frac{1}{2},j}^y\right), \tag{3.21c}$$

$$D_{xx}^+\left(\tilde{w}_{1,i}^+ f_{i,j}^x+\tilde{w}_{1,i+1}^+ H_{i+1,j}^x+\tilde{w}_{1,i+2}^+ H_{i+2,j}^x\right)+D_{xy}^+\left(\frac{\partial H_x}{\partial y}\right)^+\Big|_{x_o,y_o}$$
$$+D_{yx}^+\left(w_{1,i-\frac{1}{2}}^+ f_{i-\frac{1}{2},j}^y+w_{1,i+\frac{1}{2}}^+ H_{i+\frac{1}{2},j}^y+w_{1,i+\frac{3}{2}}^+ H_{i+\frac{3}{2},j}^y\right)$$
$$=D_{xx}^-\left(\tilde{w}_{1,i-1}^- H_{i-1,j}^x+\tilde{w}_{1,i}^- H_{i,j}^x+\tilde{w}_{1,i+1}^- f_{i+1,j}^x\right), \tag{3.21d}$$

where $w_{l,k}^-$ and $w_{l,k}^+$ are one-sided finite difference weights, respectively, for left and right subdomains. Here the subscript l represents the interpolation ($l=0$) and the first order derivative approximation ($l=1$) and k is for grid index. The finite difference weights $w_{l,k}^-$ and $w_{l,k}^+$ are for H_y, while the weights $\tilde{w}_{l,k}^-$ and $\tilde{w}_{l,k}^+$ are for H_x, since H_x and H_y are on different locations in the Yee mesh.

Only x derivatives are discretized in Eqs. (3.21a)-(3.21d), whereas three y derivatives at the interface point (x_o,y_o) remain unsolved. In order to approximate $(\partial H_y/\partial y)^+$ at (x_o,y_o), we introduce three auxiliary points (x_o,y_o), (x_o,y_{j+1}) and (x_o,y_{j+2}) on the positive side

$$\left(\frac{\partial H_y}{\partial y}\right)^+\Big|_{x_o,y_o}\approx p_{1,j}^+(H_{o,j}^y)^+ + p_{1,j}^+ H_{o,j+1}^y + p_{1,j+2}^+ H_{o,j+2}^y. \tag{3.22}$$

We note that $(H_{o,j}^y)^+$, $H_{o,j+1}^y$ and $H_{o,j+2}^y$ are evaluated on non-grid nodes. Therefore, they should be calculated by further interpolations. The grid nodes $(x_{i-1/2},y_{j+1})$, $(x_{i+1/2},y_{j+1})$, $(x_{i+3/2},y_{j+1})$ and $(x_{i-3/2},y_{j+2})$, $(x_{i-1/2},y_{j+2})$, $(x_{i+1/2},y_{j+2})$ are chosen to approximate $H_{o,j+1}^y$ and $H_{o,j+2}^y$ respectively. See Fig. 4. For evaluating $(H_{o,j}^y)^+$, one can refer it to the related grid nodes by using the following equation

$$(H_{o,j}^y)^+ = -\frac{C_x^+}{C_y^+}(H_{o,j}^x)^+ + \frac{C_y^-}{C_y^+}(H_{o,j}^y)^-$$
$$= -\frac{C_x^+}{C_y^+}(\tilde{w}_{0,i}^+ f_{i,j}^x+\tilde{w}_{0,i+1}^+ H_{i+1,j}^x+\tilde{w}_{0,i+2}^+ H_{i+2,j}^x)$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

$$+\frac{C_y^-}{C_y^+}\left(w^-_{0,i-\frac{3}{2}}H^y_{i-\frac{3}{2},j}+w^-_{0,i-\frac{1}{2}}H^y_{i-\frac{1}{2},j}+w^-_{0,i+\frac{1}{2}}f^y_{i+\frac{1}{2},j}\right). \tag{3.23}$$

Since $(\partial H_x/\partial y)^+$ is evaluated on the interface point (x_o,y_o), it can be discretized in the same manner as the approximation of $(\partial H_y/\partial y)^+$

$$\left(\frac{\partial H_x}{\partial y}\right)^+\Big|_{x_o,y_o}=\tilde{p}^+_{1,j}(H^x_{o,j})^+ +\tilde{p}^+_{1,j+\frac{1}{2}}H^x_{o,j+\frac{1}{2}}+\tilde{p}^+_{1,j+\frac{3}{2}}H^x_{o,j+\frac{3}{2}}. \tag{3.24}$$

Similarly, grid nodes $(x_i,y_{j+1/2})$, $(x_{i+1},y_{j+1/2})$, $(x_{i+2},y_{j+1/2})$ and $(x_{i-1},y_{j+3/2})$, $(x_i,y_{j+3/2})$, $(x_{i+1},y_{j+3/2})$ are chosen to interpolate $H^x_{o,j+1/2}$ and $H^x_{o,j+3/2}$, respectively. See Fig. 4. Discretization of $(H^x_{o,j})^+$ is achieved by adopting the following relation

$$\begin{aligned}(H^x_{o,j})^+ &=-\frac{D_y^+}{D_x^+}H_y^+ +\frac{D_x^-}{D_x^+}H_x^-\\ &=-\frac{D_y^+}{D_x^+}\left(w^+_{0,i-\frac{1}{2}}f^y_{i-\frac{1}{2},j}+w^+_{0,i+\frac{1}{2}}H^y_{i+\frac{1}{2},j}+w^+_{0,i+\frac{3}{2}}H^y_{i+\frac{3}{2},j}\right)\\ &\quad+\frac{D_x^-}{D_x^+}\left(\tilde{w}^-_{0,i-1}H^x_{i-1,j}+\tilde{w}^-_{0,i}H^x_{i,j}+\tilde{w}^-_{0,i+1}f^x_{i+1,j}\right).\end{aligned} \tag{3.25}$$

By combining equations from (3.21a) to (3.25) together, we obtain a system of equations which can be equivalently transferred into the following matrix form

$$\mathbf{W}\cdot\mathbf{F}=\mathbf{C}\cdot\mathbf{U}. \tag{3.26}$$

The coefficients of this matrix equation are given in Appendix C. Then the fictitious values **F** are determined as the follows

$$\mathbf{F}=\mathbf{W}^{-1}\cdot\mathbf{C}\cdot\mathbf{U}. \tag{3.27}$$

Formulation (3.27) shows that fictitious values $\mathbf{F}=[f^y_{i-1/2,j},f^y_{i+1/2,j},f^x_{i,j},f^x_{i+1,j}]^\top$ are linear combinations of the function values in **U**. In other words, they are determined by using 12 nearby H_x and H_y values. It is noted that the coefficients of fictitious values, i.e., $\mathbf{W}^{-1}\cdot\mathbf{C}$, only need to be generated once at the beginning. Then they can be re-used in all the subsequent time steps [42].

4 Discontinuous Galerkin algorithm

In this section, we will give a brief introduction to the DGTD method for solving Maxwell's equations in a 2D domain using triangle element. This domain is consisted with heterogeneous materials in different regions. For time integration, we adopt the low-storage five-stage fourth-order explicit Runge-Kutta (LSERK) scheme [13]. For spatial discretization, we will first introduce nodal triangle element and then DG method for Maxwell's equations. The detailed information can be found in related references [12,13].

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

4.1 The nodal triangle element

Suppose 2D domain $\Omega \subset R^2$ has been discretized to Ω_K with boundary $\partial\Omega$. Here Ω_K consists of K nonoverlapping triangles, i.e.,

$$\Omega \approx \Omega_K = \cup_k D_k. \tag{4.1}$$

In each element, D_k, the unknown solutions can be expressed as

$$q(\vec{x},t) \approx q_N(\vec{x},t) = \sum_{j=0}^{N_p} q(\vec{x}_j,t) L_j(\vec{x}) = \sum_{j=0}^{N_p} \hat{q}_j(t) \varphi_j(\vec{x}), \tag{4.2}$$

where $q(\vec{x}_j,t)$ is the solution at N_p grid points in triangles, $\vec{x}_j \in D_k$ and $L_j(\vec{x})$ is the 2D Lagrange polynomial on N_p grid points. Here N_p is related to polynomial order P by

$$N_p = \frac{(P+1)(P+2)}{2}. \tag{4.3}$$

Here $\varphi_j(\vec{x})$ is the orthogonal polynomial bases on triangle and $\hat{q}_j$ is the spectral coefficients for φ_j. In our previous research [38], a modal base on quadrilateral has been used and in another research [39], the nodal base has been used.

In Fig. 5, we define a mapping, $\Psi(\vec{r})$, from standard straight-sided triangle to arbitrary triangle as

$$\vec{x} = -\frac{r+s}{2}\vec{v}^1 + \frac{r+1}{2}\vec{v}^2 + \frac{s+1}{2}\vec{v}^3 = \Psi(\vec{r}), \tag{4.4}$$

where the standard triangle is defined as

$$I = \{\vec{r} = (r,s) | (r,s) \geq -1;\ r+s \leq 0\}. \tag{4.5}$$

The nodal bases for $P=4$ are shown in Fig. 6, while $P=1$ will be used to match the FDTD method in the numerical experiments. In order to construct the nodal bases L_j, we make use of the orthogonal bases φ_j through following relations

$$\vec{u} = V\hat{\vec{u}}, \quad V^T L(\vec{r}) = \tilde{\varphi}(\vec{r}), \quad V_{ij} = \varphi_j(\vec{r}_i), \tag{4.6}$$

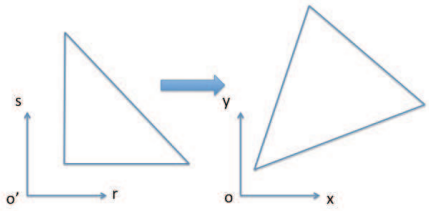


Figure 5: Triangle transformation.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

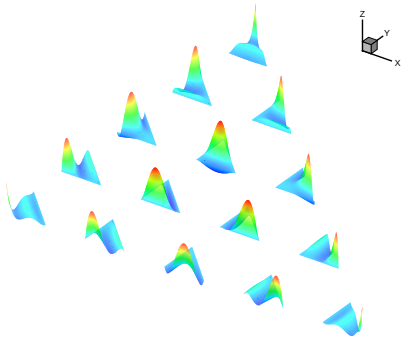


Figure 6: Nodal bases on triangle, $P=4$.

where u is any function expressed on triangle and V is the Vandermonde matrix defined in above.

$$\varphi_{ij}(\vec{r})=\sqrt{2}P_i(a)P_j^{(2i+1,0)}(b)(1-b)^i,\quad a=2\frac{1+r}{1-s}-1,\quad b=s, \tag{4.7}$$

and $P_n^{\alpha,\beta}(x)$ is the n-th order Jacobi polynomial. Using above relation, derivatives and integration can be realized accurately on triangle.

4.2 DG Method for Maxwell's equation

Let us define some notations for the DG. We refer to the interior side of the element by a superscript "-" and to the opposite side by a superscript "+". Then we have the average as

$$\{\{u\}\}=\frac{u^-+u^+}{2},\quad \{\{\vec{u}\}\}=\frac{\vec{u}^-+\vec{u}^+}{2}, \tag{4.8}$$

where u can be both a scalar and a vector. Similarly, we define the jumps along a normal, $\vec{n}$, as

$$[[u]]=\vec{n}^-u^-+\vec{n}^+u^+,\quad [[\vec{u}]]=\vec{n}^-\cdot\vec{u}^-+\vec{n}^+\cdot\vec{u}^+. \tag{4.9}$$

As shown above, $\{\{\cdot\}\}$ and $[[\cdot]]$ have different forms when dealing with a scalar or a vector.

Let us express Maxwell's equations, Eqs. (2.1a) and (2.1b), in conservation form

$$Q\frac{\partial\vec{q}}{\partial t}+\nabla\cdot\vec{F}(\vec{q})=\vec{S}, \tag{4.10}$$

where the material matrix, $Q(\vec{x})$, the state vector, $\vec{q}$ and the flux, $\vec{F}(\vec{q})$, are

$$Q(\vec{x})=\begin{bmatrix}\epsilon & 0\\ 0 & \mu\end{bmatrix},\quad \vec{q}=\begin{bmatrix}\vec{E}\\ \vec{H}\end{bmatrix},\quad \vec{F}_i(\vec{q})=\begin{bmatrix}-e_i\times\vec{H}\\ e_i\times\vec{E}\end{bmatrix},$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

with $\vec{F}(\vec{q})=[F_1(\vec{q}),F_2(\vec{q}),F_3(\vec{q})]^T$. Here e_i signifies the three Cartesian unit vectors and $\vec{S}=[S^E,S^H]^T$ represents body forces, e.g., currents and terms introduced by the scattered field formulation, Eqs. (2.1a) and (2.1b).

Suppose unknown solutions can be expressed as (4.2), we require that Eq. (4.10) be satisfied as

$$\int_D \left(Q\frac{\partial \vec{q}_N}{\partial t}+\nabla\cdot\vec{F}_N-\vec{S}_N\right)L_i(\vec{x})dx=\oint_{\partial D}\tau(\vec{x})L_i(\vec{x})\vec{n}\cdot(\vec{F}_N-\vec{F}_N^*)dx. \tag{4.11}$$

Usually $\tau(\vec{x})=1.0$ and $\vec{F}_N$ can be expressed as

$$\vec{n}\cdot\vec{F}_N=\begin{bmatrix} -\vec{n}\times\vec{H}_N \\ \vec{n}\times\vec{E}_N \end{bmatrix}.$$

4.2.1 DG numerical flux

The numerical flux is deduced from Rankine-Hugoniot condition. Suppose the eigenvalues of $Q^{-1}\nabla\cdot F$ are $c^{\pm}$, which are corresponding to the speeds of two characteristics waves. These two waves are moving in opposite directions and the states separated by them are $\vec{q}^-,\vec{q}^*,\vec{q}^+$. Then we have

$$c^+Q^+(\vec{q}^+-\vec{q}^*)=\vec{n}\cdot\vec{F}(\vec{q}^+)-\vec{n}\cdot\vec{F}(\vec{q}^*), \tag{4.12a}$$

$$-c^-Q^-(\vec{q}^*-\vec{q}^-)=\vec{n}\cdot\vec{F}(\vec{q}^*)-\vec{n}\cdot\vec{F}(\vec{q}^-). \tag{4.12b}$$

From Eqs. (4.12a) and (4.12b), the numerical flux for the DG can be deduced as

$$\vec{n}\cdot\vec{F}_N(\vec{q}^*)=\begin{cases} \dfrac{1}{2\{\{Z\}\}}(-\vec{n}\times\{\{Z\vec{H}\}\}+\vec{n}\times\vec{n}\times[\vec{E}]), \\ \dfrac{1}{2\{\{Y\}\}}(\vec{n}\times\{\{Z\vec{E}\}\}+\vec{n}\times\vec{n}\times[\vec{H}]). \end{cases} \tag{4.13}$$

Since the numerical fluxes for the electric and magnetic fields have following forms respectively,

$$\vec{n}\cdot(\vec{F}_N-\vec{F}_N^*)=\begin{bmatrix} -\vec{n}\times(\vec{H}_N-\vec{H}_N^*) \\ \vec{n}\times(\vec{E}_N-\vec{E}_N^*) \end{bmatrix},$$

it can be arrived at

$$-(\vec{n}\times\vec{H}_N-(\vec{n}\times\vec{H})_N^*)=-\frac{1}{2\{\{Z\}\}}\vec{n}\times(Z^+[\vec{H}_N]-\alpha\vec{n}\times[\vec{E}]_N), \tag{4.14a}$$

$$(\vec{n}\times\vec{E}-(\vec{n}\times\vec{E})^*)=\frac{1}{2\{\{Y\}\}}\vec{n}\times(Y^+[\vec{E}_N]+\alpha\vec{n}\times\vec{H}_N), \tag{4.14b}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

where $\alpha=0$ corresponds to central flux and $\alpha=1$ corresponds to upwind flux. Also we have $[\vec{E}_N]=\vec{E}^- -\vec{E}^+$, $[\vec{H}_N]=\vec{H}^- -\vec{H}^+$. The local impedance, $Z^\pm$ and conductance, $Y^\pm$ are

$$Z^\pm=\frac{1}{Y^\pm}=\sqrt{\frac{\mu^\pm}{\epsilon^\pm}}, \tag{4.15a}$$

$$\bar{Z}=Z^+ +Z^-=\frac{1}{\bar{Y}}=\frac{1}{Y^+ +Y^-}, \tag{4.15b}$$

$$\bar{Z}=2\{\{Z\}\}=\bar{Y}=2\{\{Y\}\}. \tag{4.15c}$$

Now returning to the semidiscrete scheme, Eq. (4.11), we have an element wise expression for the electric field components

$$\sum_{j=0}^{N}\Big(M_{ij}^{\epsilon}\frac{dE_j}{dt}-\vec{S}_{ij}\times H_j-M_{ij}S_j^E\Big)=\sum_l F_{il}\Big(-\frac{1}{\{\{Z\}\}_l}\vec{n}\times[Z_l^+[H_l]-\alpha\vec{n}\times[E_l]]\Big), \tag{4.16a}$$

$$\sum_{j=0}^{N}\Big(M_{ij}^{\epsilon}\frac{dH_j}{dt}-\vec{S}_{ij}\times E_j-M_{ij}S_j^H\Big)=\sum_l F_{il}\Big(\frac{1}{\{\{Y\}\}_l}\vec{n}\times[Y_l^+[E_l]+\alpha\vec{n}\times[H_l]]\Big). \tag{4.16b}$$

Here we have

$$M_{ij}^{\epsilon}=(L_i(\vec{x}),\epsilon(\vec{x})L_j(\vec{x}))_D,\quad M_{ij}^{\mu}=(L_i(\vec{x}),\mu(\vec{x})L_j(\vec{x}))_D, \tag{4.17}$$

as the material-scaled mass matrices and

$$M_{ij}=(L_i(\vec{x}),L_j(\vec{x}))_D,\quad S_{ij}^{\mu}=(S_{ij}^x,S_{ij}^y,S_{ij}^z)=(L_i(\vec{x}),\nabla L_j(\vec{x}))_D, \tag{4.18}$$

represent the local mass and stiffness matrices. Note that in the special case where ϵ and μ are element-wise constant, we recover $(M^\epsilon,M^\mu)=(\epsilon M,\mu M)$.

Expressing Eqs. (4.16a) and (4.16b) in fully explicit form yield

$$\begin{aligned}\frac{d\vec{E}_N}{dt}&=(M^\epsilon)^{-1}S\times\vec{H}_N+(M^\epsilon)^{-1}MS^E\\&\quad+(M^\epsilon)^{-1}F\Big(-\frac{1}{\{\{Z\}\}_l}\vec{n}\times[Z_l^+[H_N]-\alpha\vec{n}\times[E_N]]\Big)\Big|_{\partial D},\end{aligned} \tag{4.19a}$$

$$\begin{aligned}\frac{d\vec{H}_N}{dt}&=-(M^\mu)^{-1}S\times\vec{E}_N+(M^\mu)^{-1}MS^H\\&\quad+(M^\mu)^{-1}F\Big(\frac{1}{\{\{Y\}\}}\vec{n}\times[Y^+[E_N]+\alpha\vec{n}\times[H_N]]\Big)\Big|_{\partial D}.\end{aligned} \tag{4.19b}$$

Following above, in 2D case, we have the numerical flux term $\vec{F}^*$ in following form

$$\hat{n}\cdot(\vec{F}-\vec{F}^*)=\frac{1}{2}\begin{cases}\bar{Y}^{-1}(n_yn_x[H_N^y]-n_y^2[H_N^x]+n_y[E_N^z]),\\ \bar{Y}^{-1}(n_x^2[H_N^y]-n_xn_y[H_N^x]-n_x[E_N^z]),\\ \bar{Z}^{-1}(-Z^+(n_x[H_N^y]-n_y[H_N^x])+[E_N^z]),\end{cases}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

where $\vec{H}=(H^x,H^y)$ and we use the notation

$$[q]=q^- -q^+=\vec{n}\cdot[[q]]. \tag{4.20}$$

The semi-discrete scheme are

$$\frac{dH_h^x}{dt}=-D_yE_h^z+\frac{1}{2JM}\int_{\partial D^k}(\bar{Y}^{-1}(n_yn_x[H_N^y]-n_y^2[H_N^x]+n_y[E_N^z]))l(x)dx, \tag{4.21a}$$

$$\frac{dH_h^y}{dt}=D_xE_h^z+\frac{1}{2JM}\int_{\partial D^k}(\bar{Y}^{-1}(n_x^2[H_N^y]-n_xn_y[H_N^x]-n_x[E_N^z]))l(x)dx, \tag{4.21b}$$

$$\frac{dE_h^z}{dt}=D_xH_h^y-D_yH_h^x+\frac{1}{2JM}\int_{\partial D^k}(\bar{Z}^{-1}(-Z^+(n_x[H_N^y]-n_y[H_N^x])+[E_N^z]))l(x)dx. \tag{4.21c}$$

More details about the nodal based DGTD can be found in [13].

4.3 Parallel implementation

In order to handle large scale simulations, parallel computation is necessary. Due to decoupling of mass matrix for different elements, the DGTD has advantages of easy parallelization and relatively good scalability. The only communication is the exchange of numerical fluxes over element boundaries at each step. Specific arrays and mappings have been created to store the values on the opposite of each edge. MPI functions have been used for communication between processors. The mesh is created by using the software CUBIT. With the CUBIT, several million tetrahedron elements can be generated on a desktop computer. Several routines for the mesh partitioning have also been developed. The simplest one just evenly distributes the elements by its global id to all processors. A more meaningful approach is dividing the domain evenly according to the location in the domain. The publicly available software packages METIS and PARMETIS have also been used for mesh partition. In order to output data efficiently, specific parallel I/O routines have been developed in the computation. The TECPLOT software has been used for visualizing the results.

5 Experimental results

In this section, we examine the performance of the MIBTD and DGTD methods for solving discontinuous Maxwell interface problems. The stability, accuracy and convergence of two algorithms will be investigated, by considering circular interface problems in both TM and TE modes. For simplicity, a square domain $\Omega=[-1,1]\times[-1,1]$ is used in all tests. Analytical solutions are available to exactly verify numerical errors. For simplicity, the exact solutions will also be utilized for generating the initial solutions and boundary conditions in our computations. In both MIBTD and DGTD methods, the initial time and stopping time will be assigned to be $t=0$ and $t=T$ respectively. The time increment is denoted as $\Delta t=T/N_t$, with N_t being the number of time steps.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

In the MIBTD algorithm, a uniform Yee mesh is employed, in which the spacing in both x and y directions are the same, i.e., $\Delta x=\Delta y$. For simplicity, we denote $h=\Delta x=\Delta y$. Since Ω is a square domain, the grid numbers in both x and y directions are also the same and are denoted as $N=N_x=N_y$ for the on-grid component, i.e., the E_z component in the TM mode and the H_z component in the TE mode. Denoting $u_h(x_i,y_j,T)$ as the numerical approximation to a field component $u(x,y,T)$ at time T, we use the following measures to estimate L_∞ and L_2 errors of the MIBTD method

$$L_\infty=\max_{i,j=1,\cdots,N}|u(x_i,y_j,T)-u_h(x_i,y_j,T)|,$$
$$L_2=\sqrt{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}|u(x_i,y_j,T)-u_h(x_i,y_j,T)|^2}.$$

In the DGTD method, the triangle mesh is generated by the software CUBIT. For simplicity, we will also denote the spacing of the DGTD discretization as h, which is the value being specified by the user in the CUBIT to generate a mesh automatically. The length of triangular edges on the domain boundary will be determined by this h value. The total number of elements is denoted as N_s. Unless specified otherwise, the polynomial order of nodal bases is chosen as $P=1$ to match the MIBTD method. The L_∞ and L_2 errors of the DGTD method are computed as

$$L_\infty=\max|u^k(x,y,T)-u_h^k(x,y,T)|,\quad k\in[0,K-1], \tag{5.1a}$$
$$L_2=\sqrt{\frac{\sum_k\int_{D_k}(u^k(x,y,T)-u_h^k(x,y,T))^2ds}{|\Omega|}}. \tag{5.1b}$$

5.1 Transverse magnetic mode

Consider a dielectric cylinder embedded in the free space. By setting the center of the cylinder at $(0,0)$, we denote the radius to be r_0. We assign $\epsilon_1=\mu_1=1$ to be the relative parameters of vacuum while ϵ_2 and μ_2 are the relative dielectric parameters of the cylinder. The transverse magnetic (TM) electromagnetic wave is scattered by a time-harmonic incident plane unit wavelength wave of the form [3,30]

$$E_z^{\text{inc}}=e^{-i(k_1x+\omega t)}, \tag{5.2}$$

where $k_1=\omega\sqrt{\mu_1\epsilon_1}$ is the propagation constant in the free-space medium and ω is the angular frequency. Then the exact solution for such incident waves is computed by using the summed-series technique [1,3] as the follows

$$E_z(x,y,t)=E_z(r,\theta,t)=e^{-i\omega t}\begin{cases}\displaystyle\sum_{n=-\infty}^{\infty}A_nJ_n(k_2r)e^{in\theta}, & r\le r_0,\\ \displaystyle\sum_{n=-\infty}^{\infty}(i^{-n}J_n(k_1r)+B_nH_n^{(2)}(k_1r))e^{in\theta}, & r>r_0,\end{cases} \tag{5.3}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

where (r,θ) is the polar coordinate at (x,y), $k_2=\omega\sqrt{\mu_2\epsilon_2}$ is the propagation constant for the dielectric cylinder, J_n and $H_n^{(2)}$ represent, respectively, the nth order Bessel function of the first kind and the Hankel function of the second kind and the wave amplitude coefficients equal to

$$A_n=i^{-n}\frac{(k_1/\mu_1)J_n'(k_1r_0)H_n^{(2)}(k_1r_0)-(k_1/\mu_1)H_n^{(2)'}(k_1r_0)J_n(k_1r_0)}{(k_2/\mu_2)J_n'(k_2r_0)H_n^{(2)}(k_1r_0)-(k_1/\mu_1)H_n^{(2)'}(k_1r_0)J_n(k_2r_0)}, \tag{5.4a}$$

$$B_n=i^{-n}\frac{(k_1/\mu_1)J_n'(k_1r_0)J_n(k_2r_0)-(k_2/\mu_2)J_n'(k_2r_0)J_n(k_1r_0)}{(k_2/\mu_2)J_n'(k_2r_0)H_n^{(2)}(k_1r_0)-(k_1/\mu_1)H_n^{(2)'}(k_1r_0)J_n(k_2r_0)}. \tag{5.4b}$$

Then the radial component and angular component of the total magnetic field are given as

$$H_r(r,\theta,t)=e^{-i\omega t}\begin{cases}\dfrac{-i}{\omega\mu_2 r}\displaystyle\sum_{n=-\infty}^{\infty} inA_nJ_n(k_2r)e^{in\theta}, & r\le r_0,\\ \dfrac{-i}{\omega\mu_1 r}\displaystyle\sum_{n=-\infty}^{\infty} in(i^{-n}J_n(k_1r)+B_nH_n^{(2)}(k_1r))e^{in\theta}, & r>r_0,\end{cases} \tag{5.5a}$$

$$H_\theta(r,\theta,t)=e^{-i\omega t}\begin{cases}\dfrac{ik_2}{\omega\mu_2}\displaystyle\sum_{n=-\infty}^{\infty} A_nJ_n'(k_2r)e^{in\theta}, & r\le r_0,\\ \dfrac{ik_1}{\omega\mu_1}\displaystyle\sum_{n=-\infty}^{\infty} (i^{-n}J_n'(k_1r)+B_nH_n^{(2)'}(k_1r))e^{in\theta}, & r>r_0.\end{cases} \tag{5.5b}$$

By using the coordinate transformation, exact solutions for H_x and H_y are determined as the follows

$$H_x=\cos\theta H_r-\sin\theta H_\theta,\quad H_y=\sin\theta H_r+\cos\theta H_\theta. \tag{5.6}$$

In the following tests, we assign the radius of the cylinder to be $r_0=0.4$ and angular frequency to be $\omega=2\pi$. Two cases will be studied. In the first case, we take $\epsilon_2=10$ and $\mu_2=1$. Since $\mu_1=\mu_2$, we have that all three components E_z, H_x and H_y being continuous. In the second case, we choose $\epsilon_2=\mu_2=10$. Then the magnetic components H_x and H_y are discontinuous across the interface Γ, while E_z is still continuous. We note that the Bessel functions are singular at the origin $(0,0)$. To avoid possible numerical problems, we need to make sure that the origin is not sampled in our grids for the MIBTD and DGTD methods. For the MIBTD method, this can be done by choosing the mesh size N as an even integer, whereas a similar constrain can be implemented in the DGTD method too.

We first study the stability of the MIBTD and DGTD methods with a large enough stopping time $T=10$. It is well known that the Courant-Friedrichs-Lewy (CFL) stability condition for the FDTD algorithm can be expressed as [42]

$$\Delta t\le C_m\frac{h}{v_{\max}\sqrt{d}}, \tag{5.7}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Table 1: Critical CFL number of the MIBTD algorithm for the TM mode.

N	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$		Case 2: $(\epsilon_2,\mu_2)=(10,10)$	
		Critical N_t	C_m	Critical N_t	C_m
20	1.05E-1	101	1.3302	99	1.3571
40	5.13E-2	197	1.3999	195	1.4142
80	2.53E-2	396	1.4106	394	1.4178
160	1.26E-2	825	1.3628	816	1.3778
320	6.27E-3	1654	1.3638	1956	1.1532

where $d=2$ is the dimension number and the maximum velocity of the propagating wave is defined as

$$v_{\max}=\max\left\{\frac{1}{\sqrt{\mu_1\epsilon_1}},\frac{1}{\sqrt{\mu_2\epsilon_2}}\right\}.$$

Since we have $\mu_1=\epsilon_1=1$ in both cases, the maximum velocity is fixed to be $v_{\max}=1$. The value of the CFL constant C_m is determined by both the spatial and temporal discretizations. The procedure to numerically identify this CFL constant has been detailed in [30]. Essentially, we need to search for a critical number of time steps, N_t, which just remains stable. The numerically detected CFL value C_m and critical N_t of the MIBTD algorithm are reported in Table 1. For the smooth problem, i.e., Case 1, C_m goes to a number around 1.36 as h approaches zero. The CFL number C_m takes a smaller number when the solutions are discontinuous in Case 2.

For the DGTD method, the stability condition is known to be [12]

$$\Delta t\le C_d\cdot\min\{\sqrt{\epsilon\mu}|\chi|^{-1}\} \tag{5.8}$$

with $\sqrt{\epsilon\mu}$ reflecting the modified local speed of light due to materials and

$$\chi=\frac{|\nabla\xi|}{\Delta\xi}+\frac{|\nabla\eta|}{\Delta\eta}+\frac{|\nabla\zeta|}{\Delta\zeta}. \tag{5.9}$$

Here $|\cdot|$ refers to the absolute value of each and of the vector components, i.e., $|\nabla\xi|=[|\xi_x|,|\xi_y|,|\xi_z|]^T$. Hence, χ provides a measure of the local grid distortion as a consequence of the mapping, Ψ, of I into D and $(\Delta\xi,\Delta\eta,\Delta\zeta)$ measures the axial distance separating neighboring nodal points in I. In Table 2, the critical N_t and the CFL number C_d of the DGTD algorithm are reported. Unlike the MIBTD method, the CFL number of the DGTD method is the same for both Case 1 and Case 2, when the same triangular mesh is used. Thus, only one set of data is shown in Table 2. It is seen that C_d goes to around 1.15 when h goes to zero. This asymptotic value is very close to that of the MIBTD method in Case 2, but the DGTD method usually needs a larger critical N_t.

We next investigate the accuracy of MIBTD and DGTD methods with a stopping time $T=1$. For both MIBTD and DGTD methods, we choose $C_m=C_d=0.7$ to ensure stability. For each grid, the Δt values can be correspondingly determined. The results of MIBTD and DGTD methods are reported, respectively, in the Table 3 and Table 4. Here only the

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Table 2: Critical CFL number of the DGTD algorithm for both TM and TE modes.

N_s	h	Critical N_t	C_d
886	1.00E-1	305	1.045
3506	5.00E-2	650	1.107
13884	2.50E-2	1281	1.114
54708	1.25E-2	2557	1.138
220396	6.25E-3	5085	1.152

Table 3: Numerical convergence test of the MIBTD algorithm for the TM mode.

N	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$				Case 2: $(\epsilon_2,\mu_2)=(10,10)$			
		L_∞ error	Order	L_2 error	Order	L_∞ error	Order	L_2 error	Order
20	1.05E-1	9.63E-1		1.89E-1		3.20E-0		6.23E-1	
40	5.13E-2	2.67E-1	1.85	5.32E-2	1.83	3.62E-0	-0.18	4.77E-1	0.39
80	2.53E-2	6.71E-2	1.99	1.34E-2	1.99	1.82E-0	0.99	1.90E-1	1.33
160	1.26E-2	1.67E-2	2.00	3.30E-3	2.02	5.73E-1	1.67	5.17E-2	1.88
320	6.27E-3	4.20E-3	1.99	8.24E-4	2.00	1.50E-1	1.94	1.31E-2	1.98

Table 4: Numerical convergence test of the DGTD algorithm for the TM mode.

N_s	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$				Case 2: $(\epsilon_2,\mu_2)=(10,10)$			
		L_∞ error	Order	L_2 error	Order	L_∞ error	Order	L_2 error	Order
886	1.00E-1	4.93E-1		1.30E-1		3.99E-0		7.32E-1	
3506	5.00E-2	1.22E-1	2.02	2.83E-2	2.20	4.22E-0	-0.08	4.18E-1	0.81
13884	2.50E-2	2.84E-2	2.10	6.70E-3	2.08	1.07E-0	1.98	1.15E-1	1.86
54708	1.25E-2	9.07E-3	1.65	1.52E-3	2.14	2.51E-1	2.09	2.58E-2	2.16
220396	6.25E-3	2.36E-3	1.94	4.09E-4	1.89	6.66E-2	1.91	6.32E-3	2.03

errors in the E_z component are listed in both L_∞ and L_2 norms. The numerical errors of all three components E_z, H_y and H_x are plotted in Fig. 7. Successive mesh refinements are considered, so that we can calculate the numerical orders after each refinement. It can be seen from tables and figures that the DGTD method are usually slightly more accurate than the MIBTD method. Both MIBTD and DGTD methods deliver a second order convergence in these studies, except when grids are too coarse in beginning refinements of the Case 2. This is because the wavenumber k_2 is too large with $\epsilon_2=\mu_2=10$. When one refines the mesh, the DGTD starts to converge, while the MIBTD follows in a later refinement. Overall, once the convergence takes place, the convergence rate of the MIBTD method is slightly more uniform than the DGTD method, especially in terms of the L_∞ norm.

The MIBTD algorithm is by design a second order scheme, while the DGTD algorithm has the potential to deliver a higher order of accuracy. We have explored this by considering the polynomial order of nodal bases being $P=2$ in the DGTD computations of the present TM test problems. The accuracies and convergence rates of the DGTD algorithm for the E_z component is reported in Table 5. Here only the TM Case 1 is shown, since the results of the other case are similarly. In comparing the present results with the previous

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

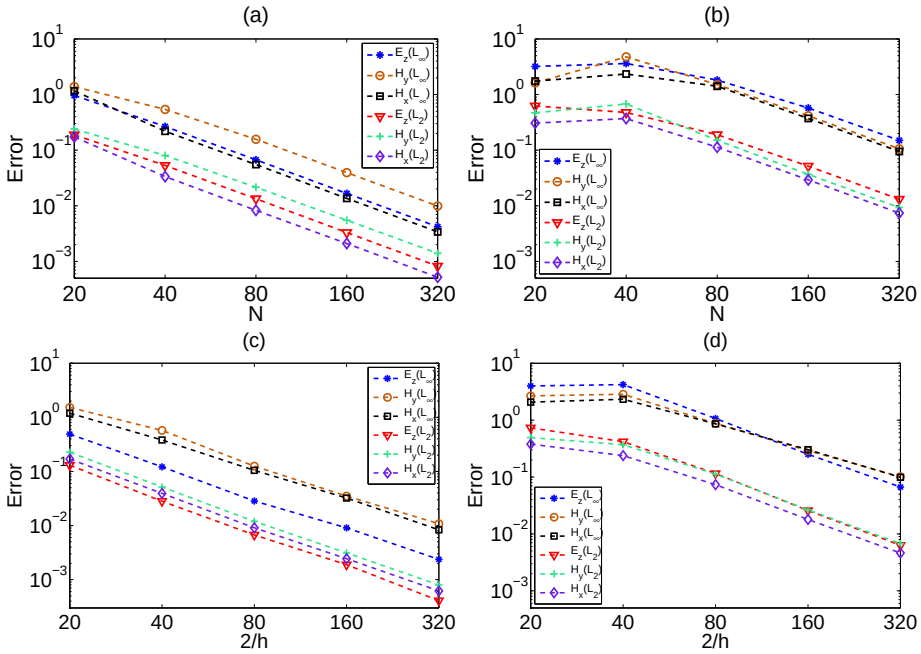


Figure 7: Numerical convergence tests for the TM mode. (a) MIBTD scheme, Case 1 $(\epsilon_2,\mu_2)=(10,1)$; (b) MIBTD scheme, Case 2 $(\epsilon_2,\mu_2)=(10,10)$; (c) DGTD scheme, Case 1 $(\epsilon_2,\mu_2)=(10,1)$; (d) DGTD scheme, Case 2 $(\epsilon_2,\mu_2)=(10,10)$.

ones based on $P=1$, it is found that the DGTD method with $P=2$ indeed yields more accurate solutions. However, the convergence rates in both L_2 and L_∞ norms do not reach the third order, even though they are slightly over the second order. In other words, the optimal convergence is not achieved in the present DGTD computations with $P=2$. As noted in [16], in which a DG method is applied to solve the elliptic interface problem, the essential reason for the order reduction of the DG method is due to the geometrical approximation of curved interfaces by regular triangular elements. Superparametric elements are utilized in [16] to recover the optimal convergences of the DG method. The implementation of such superparametric elements in the present nodal based DG method is of great interests and will be explored in the future.

Table 5: Numerical convergence test of the DGTD algorithm with $P=2$ for the TM mode.

N_e	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$			
		L_∞ error	Order	L_2 error	Order
234	2.00E-1	4.59E-1		1.03E-1	
886	1.00E-1	1.79E-1	1.83	1.77E-2	2.54
3506	5.00E-2	3.85E-2	2.22	3.80E-2	2.22
13884	2.50E-2	5.99E-3	2.68	9.87E-4	1.94
54708	1.25E-2	1.27E-3	2.24	2.49E-4	1.99

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Table 6: The execution time in seconds for TM Case 1 with different CPUs (1, 4, 16 and 64).

1 CPU	4 CPUs		16 CPUs		64 CPUs	
Time	Time	Efficiency	Time	Efficiency	Time	Efficiency
7320.55	1665.555	1.098	448.22	0.929	191.37	0.585

The DGTD method is implemented in a parallel environment. To illustrate the CPU time acceleration, we consider the Case 1 of the TM mode with $h=0.025$. The efficiency gain in using 4, 16 and 64 processors is reported in Table 6. Even though the efficiency becomes smaller when the number of processors increases, the CPU time indeed becomes significantly shorter in the parallel computing.

5.2 Transverse electric mode

We next consider the transverse electric (TE) mode. The same circular interface and dielectric coefficients as in the TM mode are employed for the TE mode. The incident plane wave is assumed for H_z

$$H_z^{\text{inc}}=e^{-i(k_1x+\omega t)}. \tag{5.10}$$

By using the summed-series technique [1,3] again, the exact solution for H_z field is given as the follows

$$H_z(x,y,t)=H_z(r,\theta,t)=e^{-i\omega t}\begin{cases}\displaystyle\sum_{n=-\infty}^{\infty}C_nJ_n(k_2r)e^{in\theta}, & r\le r_0,\\ \displaystyle\sum_{n=-\infty}^{\infty}(i^{-n}J_n(k_1r)+D_nH_n^{(2)}(k_1r))e^{in\theta}, & r>r_0,\end{cases} \tag{5.11}$$

with the wave amplitude coefficients

$$C_n=i^{-n}\frac{(k_1/\epsilon_1)J_n'(k_1r_0)H_n^{(2)}(k_1r_0)-(k_1/\epsilon_1)H_n^{(2)'}(k_1r_0)J_n(k_1r_0)}{(k_2/\epsilon_2)J_n'(k_2r_0)H_n^{(2)}(k_1r_0)-(k_1/\epsilon_1)H_n^{(2)'}(k_1r_0)J_n(k_2r_0)}, \tag{5.12a}$$

$$D_n=i^{-n}\frac{(k_1/\epsilon_1)J_n'(k_1r_0)J_n(k_2r_0)-(k_2/\epsilon_2)J_n'(k_2r_0)J_n(k_1r_0)}{(k_2/\epsilon_2)J_n'(k_2r_0)H_n^{(2)}(k_1r_0)-(k_1/\epsilon_1)H_n^{(2)'}(k_1r_0)J_n(k_2r_0)}. \tag{5.12b}$$

Then the radial component and angular component of the total electric field are given as

$$E_r(r,\theta,t)=e^{-i\omega t}\begin{cases}\dfrac{i}{\omega\epsilon_2 r}\displaystyle\sum_{n=-\infty}^{\infty}inC_nJ_n(k_2r)e^{in\theta}, & r\le r_0,\\ \dfrac{i}{\omega\epsilon_1 r}\displaystyle\sum_{n=-\infty}^{\infty}in(i^{-n}J_n(k_1r)+D_nH_n^{(2)}(k_1r))e^{in\theta}, & r>r_0,\end{cases} \tag{5.13a}$$

$$E_\theta(r,\theta,t)=e^{-i\omega t}\begin{cases}\dfrac{-ik_2}{\omega\epsilon_2}\displaystyle\sum_{n=-\infty}^{\infty}C_nJ_n'(k_2r)e^{in\theta}, & r\le r_0,\\ \dfrac{-ik_1}{\omega\epsilon_1}\displaystyle\sum_{n=-\infty}^{\infty}(i^{-n}J_n'(k_1r)+D_nH_n^{(2)'}(k_1r))e^{in\theta}, & r>r_0.\end{cases} \tag{5.13b}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Table 7: Critical CFL number of the MIBTD algorithm for the TE mode.

N	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$		Case 2: $(\epsilon_2,\mu_2)=(10,10)$	
		Critical N_t	C	Critical N_t	C
20	1.05E-1	112	1.1956	99	1.3571
40	5.13E-2	206	1.3387	195	1.4142
80	2.53E-2	418	1.3364	394	1.4178
160	1.26E-2	904	1.2437	816	1.3778
320	6.27E-3	1826	1.2346	1956	1.1532

Table 8: Numerical convergence test of the MIBTD algorithm for the TE mode.

N	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$				Case 2: $(\epsilon_2,\mu_2)=(10,10)$			
		L_∞ error	Order	L_2 error	Order	L_∞ error	Order	L_2 error	Order
20	1.05E-1	2.73E-0		4.16E-1		3.20E-0		6.23E-1	
40	5.13E-2	1.07E-0	1.35	1.29E-1	1.60	3.62E-0	-0.18	4.77E-1	0.39
80	2.53E-2	3.14E-1	1.77	3.32E-2	1.96	1.82E-0	0.99	1.90E-1	1.33
160	1.26E-2	7.84E-2	2.00	8.30E-3	2.00	5.73E-1	1.67	5.17E-2	1.88
320	6.27E-2	2.05E-2	1.94	2.10E-3	1.98	1.50E-1	1.94	1.31E-2	1.98

Table 9: Numerical convergence test of the DGTD algorithm for the TE mode.

N_e	h	Case 1: $(\epsilon_2,\mu_2)=(10,1)$				Case 2: $(\epsilon_2,\mu_2)=(10,10)$			
		L_∞ error	Order	L_2 error	Order	L_∞ error	Order	L_2 error	Order
886	1.00E-1	2.08E-0		2.59E-1		3.99E-0		7.32E-1	
3506	5.00E-1	4.26E-1	2.29	5.09E-2	2.35	4.22E-0	-0.08	4.17E-1	0.81
13884	2.50E-2	1.15E-1	1.90	1.12E-2	2.18	1.07E-1	1.98	1.15E-1	1.86
54708	1.25E-2	3.16E-2	1.86	3.10E-3	1.85	2.51E-1	2.09	2.58E-2	2.16
220396	6.25E-3	8.75E-3	1.85	7.43E-4	2.06	6.66E-2	1.91	6.32E-3	2.03

By using the coordinate transformation, exact solutions for E_x and E_y are determined as the follows

$$E_x=\cos\theta E_r-\sin\theta E_\theta,\quad E_y=\sin\theta E_r+\cos\theta E_\theta. \tag{5.14}$$

The physical parameters of the present TE mode are chosen as in the TM mode, i.e., $r_0=0.4$ and $\omega=2\pi$. We also study two cases. For the first case, we also have $\epsilon_2=10$ and $\mu_2=1$. Nevertheless, unlike the TM mode, the present Maxwell interface problem is discontinuous. In particular, both E_x and E_y are discontinuous, while H_z is still continuous. For the second case, we also choose $\epsilon_2=\mu_2=10$. It is interesting to note that the present TE case is actually identical to the second case of the TM mode, by simply exchanging (H_z,E_x,E_y) with (E_z,H_x,H_y).

We start by checking the stability of the MIBTD and DGTD methods. The stability conditions of the DGTD method remain unchanged for the TE mode, see Table 2. For the MIBTD method, the critical CFL numbers based on a stopping time $T=10$ are reported in Table 7. Comparing with the TM results, the CFL numbers become slightly smaller in the Case 1. For the Case 2, the CFL numbers are actually identical, because the physical problem is essentially the same.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

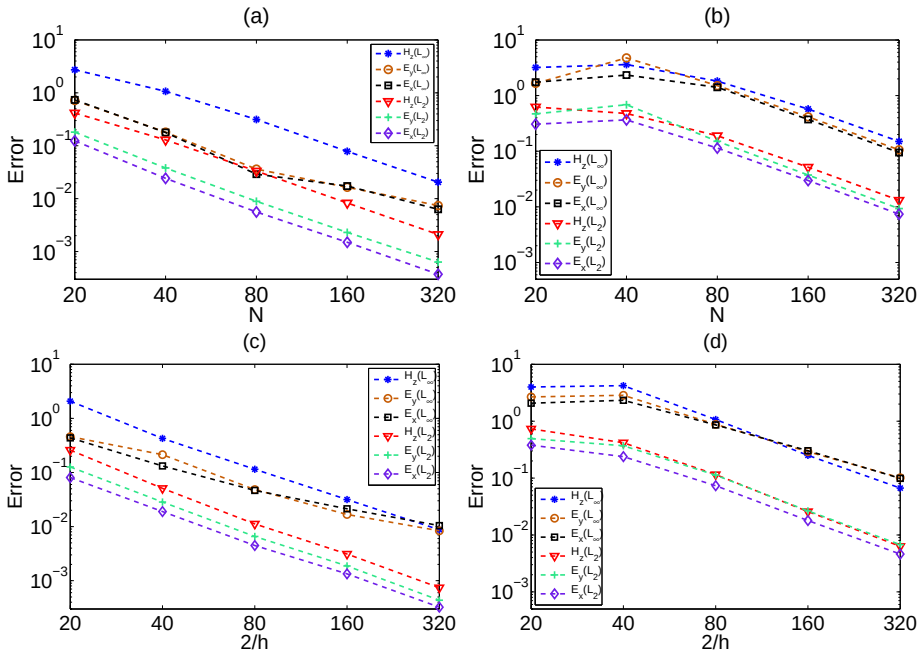


Figure 8: Numerical convergence tests for the TE mode. (a) MIBTD scheme, Case 1 $(\epsilon_2,\mu_2)=(10,1)$; (b) MIBTD scheme, Case 2 $(\epsilon_2,\mu_2)=(10,10)$; (c) DGTD scheme, Case 1 $(\epsilon_2,\mu_2)=(10,1)$; (d) DGTD scheme, Case 2 $(\epsilon_2,\mu_2)=(10,10)$.

By taking $T=1$ and $C_m=C_d=0.7$, the L_2 and L_∞ errors in the H_z component of both methods are given in Table 8 and Table 9, respectively. For a visual comparison, we plot all numerical errors of all components consisting of H_z, E_y and E_x in Fig. 8. For the MIBTD method, the convergence begins only when a sufficiently fine mesh is used, due to the impact of discontinuous solution and large wavenumber. The second order convergence is achieved on fine meshes. Moreover, the MIBTD results for Case 2 are identical to those of TM mode Case 2, due to the problem equivalence. For the DGTD method, the convergence takes place at an earlier stage. But the DGTD results for Case 2 are different from those of TM mode Case 2. Moreover, the convergence rate of the DGTD method is slightly less uniform than that of the MIBTD method, especially in terms of L_∞ norm in the Case 1. In summary, given the difficulty level of this discontinuous Maxwell interface problem, it is encouraging to see that both methods yield second order of accuracy.

6 Conclusions

In this paper, a comparative study of two time domain Maxwell solvers is conducted. The focus of this work is on the electromagnetic interface problems with discontinuous electric and/or magnetic field components, for which the conventional solvers without

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

special material interface treatments suffer a serious accuracy reduction near the interfaces. Circular interface problems with analytical solutions for both TM and TE modes are employed in this paper to benchmark the performance of a collocation approach, i.e., the MIBTD method and a Galerkin approach, i.e., the DGTD method. In the collocation approach, a novel MIB interface scheme is proposed for solving discontinuous Maxwell interface problems, in which a proper handling of discontinuous and staggered field components is offered. The physical jump conditions describing the regularity changes in wave solutions across the interface are rigorously enforced in the MIBTD scheme so that the accuracy is restored. In the Galerkin approach, the jumps of solutions across the interface are captured via a proper numerical flux, which then can be preserved in the time integration. This enables the DGTD method to accurately resolve the discontinuous Maxwell interface problems. Numerical experiments are carried out to investigate the stability, accuracy and convergence of both MIBTD and DGTD methods.

In comparison, it is interesting to note that each Maxwell solver has its own strength and weakness.

- Based on Runge-Kutta time stepping schemes, both methods are conditionally stable with similar CFL numbers. The stability constraint of the DGTD method is more uniform, while that of the MIBTD method changes slightly for the TM and TE modes and when different dielectric coefficients are used.
- Built on a parallel computing environment, the present DGTD method is surely faster than the MIBTD method.
- Both MIBTD and DGTD methods achieve a second order of convergence in numerical tests. Based on a similarly mesh spacing, the DGTD method is slightly more accurate than the MIBTD method, especially when a high wavenumber is involved, while the convergence rate of the MIBTD method is slightly more uniform, especially in terms of the L_∞ norm.
- The DGTD algorithm is very robust, in the sense that a very minor change is needed when it is applied to solve TM mode or TE mode and an interface problem with discontinuous or continuous solutions. On the contrary, a significant change in the MIBTD method is required in these scenarios. In particular, in the existing MITBD methods, the electromagnetic fields are assumed to be continuous so that the previous methods simply cannot tackle the present discontinuous Maxwell interface problem. A novel MIB scheme with a considerable improvement is thus developed in this work.

Appendix A: Jump condition construction for the electric field

For seeking the first order jump conditions of all fields in the Cartesian grid, one can use first derivative operator in (3.5). Then the first order jump conditions of E_z component in

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

the Cartesian grid arise as follows

$$0=\Big[\frac{1}{\mu}\frac{\partial E_z}{\partial n}\Big]=\cos\theta\frac{1}{\mu^+}\Big(\frac{\partial E_z}{\partial x}v\Big)^+ +\sin\theta\frac{1}{\mu^+}\Big(\frac{\partial E_z}{\partial y}\Big)^+ -\cos\theta\frac{1}{\mu^-}\Big(\frac{\partial E_z}{\partial x}\Big)^- -\sin\theta\frac{1}{\mu^-}\Big(\frac{\partial E_z}{\partial y}\Big)^-, \quad \text{(A.1a)}$$

$$0=\Big[\frac{\partial E_z}{\partial \tau}\Big]=-\sin\theta\Big(\frac{\partial E_z}{\partial x}\Big)^+ +\cos\theta\Big(\frac{\partial E_z}{\partial y}\Big)^+ +\sin\theta\Big(\frac{\partial E_z}{\partial x}\Big)^- -\cos\theta\Big(\frac{\partial E_z}{\partial y}\Big)^-. \quad \text{(A.1b)}$$

If the extrapolation is conducted from the positive side, one will cancel $(\partial E_z/\partial y)^-$ from (A.1a) and (A.1b) to achieve (3.18a), where the coefficients are given as

$$A_x^+=\frac{1}{\mu^+}\cos\theta+\frac{1}{\mu^-}\tan\theta\sin\theta,\quad A_y^+=\frac{1}{\mu^+}\sin\theta-\frac{1}{\mu^-}\sin\theta,\quad A_x^-=\frac{1}{\mu^-}\cos\theta+\frac{1}{\mu^-}\tan\theta\sin\theta,$$

$$B_x^+=\Big(\frac{1}{\mu^+}-\frac{1}{\mu^-}\Big)\cos\theta,\quad B_y^+=\frac{1}{\mu^-}\cos\theta\cot\theta+\frac{1}{\mu^+}\sin\theta,\quad B_y^-=\frac{1}{\mu^-}\Big(\cos\theta\cot\theta+\sin\theta\Big).$$

Another first order jump condition for E_z (3.18b) can be derived by canceling $(\partial E_z/\partial x)^-$ from (A.1a) and (A.1b).

Appendix B: Jump condition construction for the magnetic fields

To construct the first order jump conditions for the magnetic fields, we again apply the operator in (3.5) to conditions (3.9) and (3.12) to get

$$0=\Big[\frac{\partial(\mu H_n)}{\partial n}\Big]+\Big[\frac{\partial(\mu H_\tau)}{\partial \tau}\Big]=\Big[\frac{\partial(\mu H_x)}{\partial x}\Big]+\Big[\frac{\partial(\mu H_y)}{\partial y}\Big], \quad \text{(B.2a)}$$

$$0=\Big[\frac{\partial(\mu H_\tau)}{\partial \tau}\Big]=\sin^2\theta\Big[\frac{\partial(\mu H_x)}{\partial x}\Big]-\cos\theta\sin\theta\Big[\frac{\partial(\mu H_x)}{\partial y}\Big]-\sin\theta\cos\theta\Big[\frac{\partial(\mu H_y)}{\partial x}\Big]$$
$$+\cos^2\theta\Big[\frac{\partial(\mu H_y)}{\partial y}\Big], \quad \text{(B.2b)}$$

$$0=\Big[\frac{\partial(\mu H_n)}{\partial \tau}\Big]=-\sin\theta\cos\theta\Big[\frac{\partial(\mu H_x)}{\partial x}\Big]+\cos^2\theta\Big[\frac{\partial(\mu H_x)}{\partial y}\Big]-\sin^2\theta\Big[\frac{\partial(\mu H_y)}{\partial x}\Big]$$
$$+\cos\theta\sin\theta\Big[\frac{\partial(\mu H_y)}{\partial y}\Big]. \quad \text{(B.2c)}$$

By solving four equations (3.19), (B.2a), (B.2b) and (B.2c), we attain four simplified jump conditions

$$\Big[\frac{\partial(\mu H_y)}{\partial x}\Big]=\cos^2\theta C_0\Big\{\Big(\frac{\partial H_y}{\partial x}\Big)^+-\Big(\frac{\partial H_x}{\partial y}\Big)^+\Big\}-\sin(2\theta)\Big[\frac{\partial(\mu H_\tau)}{\partial \tau}\Big], \quad \text{(B.3a)}$$

$$\Big[\frac{\partial(\mu H_x)}{\partial x}\Big]=-\frac{\sin(2\theta)}{2}C_0\Big\{\Big(\frac{\partial H_y}{\partial x}\Big)^+-\Big(\frac{\partial H_x}{\partial y}\Big)^+\Big\}-\cos(2\theta)\Big[\frac{\partial(\mu H_\tau)}{\partial \tau}\Big], \quad \text{(B.3b)}$$

$$\Big[\frac{\partial(\mu H_y)}{\partial y}\Big]=\frac{\sin(2\theta)}{2}C_0\Big\{\Big(\frac{\partial H_y}{\partial x}\Big)^+-\Big(\frac{\partial H_x}{\partial y}\Big)^+\Big\}+\cos(2\theta)\Big[\frac{\partial(\mu H_\tau)}{\partial \tau}\Big], \quad \text{(B.3c)}$$

$$\Big[\frac{\partial(\mu H_x)}{\partial y}\Big]=-\sin^2\theta C_0\Big\{\Big(\frac{\partial H_y}{\partial x}\Big)^+-\Big(\frac{\partial H_x}{\partial y}\Big)^+\Big\}-\sin(2\theta)\Big[\frac{\partial(\mu H_\tau)}{\partial \tau}\Big], \quad \text{(B.3d)}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

where

$$C_0=\frac{\mu^+\epsilon^+-\mu^-\epsilon^-}{\epsilon^+}.$$

Now note that term $[\partial(\mu H_\tau)/\partial\tau]$ remains unknown, but can be further simplified as follows

$$\begin{aligned}\left[\frac{\partial(\mu H_\tau)}{\partial\tau}\right]&=(\mu^+-\mu^-)\left(\frac{\partial H_\tau}{\partial\tau}\right)^+ +\mu^-\left[\frac{\partial H_\tau}{\partial\tau}\right]\\&=(\mu^+-\mu^-)\left(\sin^2\theta\left(\frac{\partial H_x}{\partial x}\right)^+-\cos\theta\sin\theta\left(\frac{\partial H_x}{\partial y}\right)^+-\sin\theta\cos\theta\left(\frac{\partial H_y}{\partial x}\right)^+\right.\\&\quad\left.+\cos^2\theta\left(\frac{\partial H_y}{\partial y}\right)^+\right).\end{aligned}\tag{B.4}$$

Note that $[\partial H_\tau/\partial\tau]=0$. So it is dropped out in the derivation.

Since μ is a piecewise constant across the interface, Eq. (3.11) can be deduced to another equation

$$\frac{\partial H_x}{\partial x}=-\frac{\partial H_y}{\partial y},\tag{B.5}$$

for either positive or negative side, but not the both sides. One combines (B.3a), (B.4) and (B.5) to achieve (3.20a). Another three jump conditions (3.20b), (3.20c) and (3.20d) for these components can be obtained in the same manner. The coefficients involved in these four equations are given as

$$\begin{aligned}&C^+_{xy}=-\frac{\sin^2(2\theta)}{2}(\mu^+-\mu^-)+\cos^2\theta C_0, && C^+_{yx}=\mu^+-\frac{\sin^2(2\theta)}{2}(\mu^+-\mu^-)-\cos^2\theta C_0,\\&C^+_{yy}=\frac{\sin(4\theta)}{2}(\mu^+-\mu^-), && C^-_{yx}=\mu^-,\\&D^+_{xx}=\mu^+-\cos^2(2\theta)(\mu^+-\mu^-), && D^+_{xy}=-\frac{\sin(4\theta)}{4}(\mu^+-\mu^-)-\frac{\sin(2\theta)}{2}C_0,\\&D^+_{yx}=-\frac{\sin(4\theta)}{4}(\mu^+-\mu^-)+\frac{\sin(2\theta)}{2}C_0, && D^-_{xx}=\mu^-,\\&\hat{C}^+_{xy}=\frac{\sin(4\theta)}{4}(\mu^+-\mu^-)+\frac{\sin(2\theta)}{2}C_0, && \hat{C}^+_{yx}=\frac{\sin(4\theta)}{4}(\mu^+-\mu^-)-\frac{\sin(2\theta)}{2}C_0,\\&\hat{C}^+_{yy}=\mu^+-\cos^2(2\theta)(\mu^+-\mu^-), && \hat{C}^-_{yy}=\mu^-,\\&\hat{D}^+_{xx}=-(\mu^+-\mu^-)\frac{\sin(4\theta)}{2}, && \hat{D}^+_{xy}=\mu^+-\frac{\sin^2(2\theta)}{2}(\mu^+-\mu^-)-\sin^2\theta C_0,\\&\hat{D}^+_{yx}=-\frac{\sin^2(2\theta)}{2}(\mu^+-\mu^-)+\sin^2\theta C_0, && \hat{D}^-_{xy}=\mu^- \quad\text{and}\quad C_0=\frac{\mu^+\epsilon^+-\mu^-\epsilon^-}{\epsilon^+}.\end{aligned}$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Appendix C: MIB matrix coefficients

The entries involved in the matrix equation (3.26) are given as follows.

$$\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3,\mathbf{w}_4,]^\top\in\mathbb{R}^{4\times4},\quad \mathbf{F}=\left[f^y_{i-\frac12,j},f^y_{i+\frac12,j},f^x_{i,j},f^x_{i+1,j}\right]^\top\in\mathbb{R}^{4\times1},$$
$$\mathbf{C}=[\mathbf{c}_1,\mathbf{c}_2,\mathbf{c}_3,\mathbf{c}_4]^\top\in\mathbb{R}^{4\times12},$$
$$\mathbf{U}=\Big[H^y_{i-\frac32,j},H^y_{i-\frac12,j},H^y_{i+\frac12,j},H^y_{i+\frac32,j},H^x_{i-1,j},H^x_{i,j},H^x_{i+1,j},H^x_{i+2,j},H^y_{o,j+1},H^y_{o,j+2},$$
$$H^x_{o,j+\frac12},H^x_{o,j+\frac32}\Big]^\top\in\mathbb{R}^{12\times1},$$

where vectors

$$\mathbf{w}_1^\top=\Big(C_y^+w^+_{0,i-\frac12},C_y^-w^-_{0,i+\frac32},-C_x^+\tilde w^+_{0,i},0\Big),\quad \mathbf{w}_2^\top=\Big(-D_y^+w^+_{0,i-\frac12},0,-D_x^+\tilde w^+_{0,i},D_x^-\tilde w^-_{0,i+1}\Big),$$
$$\mathbf{w}_3^\top=\Big(-C^+_{yx}w^+_{1,i-\frac12}+\frac{D_y^+}{D_x^+}C^+_{xy}\tilde p^+_{1,j}w^+_{0,i-\frac12},C^-_{yx}w^-_{1,i+\frac32}-C^+_{yy}p^+_{1,j}\frac{C_y^-}{C_y^+}w^-_{0,i+\frac32},$$
$$-C^+_{xx}\tilde w_{1,i}+\frac{C_x^+}{C_y^+}C^+_{yy}p^+_{1,j}\tilde w^+_{0,i},-C^+_{xy}\tilde p^+_{1,j}\frac{D_x^-}{D_x^+}\tilde w^-_{0,i+1}\Big),$$
$$\mathbf{w}_4^\top=\Big(-D^+_{yx}w_{1,i-\frac12}+D^+_{xy}\tilde p^+_{1,j}\frac{D_y^+}{D_x^+}w^+_{0,i},-D^+_{yy}p^+_{1,j}\frac{C_y^-}{C_y^+}w^-_{0,i+\frac12},-D^+_{xx}\tilde w^+_{1,i}+D^+_{yy}p^+_{1,j}\frac{C_x^+}{C_y^+}\tilde w^+_{0,i},$$
$$D^-_{xx}\tilde w^-_{1,i+1}-D^+_{xy}\tilde p^+_{1,j}\frac{D_x^-}{D_x^+}\tilde w^-_{0,i+1}\Big),$$
$$\mathbf{c_1}^\top=\Big(-C_y^-w^-_{0,i-\frac32},-C_y^-w^-_{0,i-\frac12},C_y^+w^+_{0,i+\frac12}C_y^+w^+_{0,i+\frac32},0,0,C_x^+\tilde w^+_{0,i+1},C_x^+\tilde w^+_{0,i+2},0,0,0,0\Big),$$
$$\mathbf{c_2}^\top=\Big(0,0,D_y^+w^+_{0,i+\frac12},D_y^+w^+_{0,i+\frac32},-D_x^-\tilde w^-_{0,i-1},-D_x^-\tilde w^-_{0,i},D_x^+\tilde w^+_{0,i+\frac12},D_x^+\tilde w^+_{0,i+\frac32},0,0,0,0\Big),$$
$$\mathbf{c_3}^\top=\Big(-C^-_{yx}w^-_{1,i-\frac32}+C^+_{yy}p^+_{1,j}\frac{C_y^-}{C_y^+}w^-_{0,i-\frac32},-C^-_{yx}w^-_{1,i-\frac12}+C^+_{yy}p^+_{1,j}\frac{C_y^-}{C_y^+}w^-_{0,i-\frac12},$$
$$C^+_{yx}w^+_{1,i+\frac12}-\frac{D_y^+}{D_x^+}C^+_{xy}\tilde p^+_{1,j}w^+_{0,i+\frac12},C^+_{yx}w^+_{1,i+\frac32}-\frac{D_y^+}{D_x^+}C^+_{xy}\tilde p^+_{1,j}w^+_{0,i+\frac32},C^+_{xy}\tilde p^+_{1,j}\frac{D_x^-}{D_x^+}\tilde w^-_{0,i-1},$$
$$C^+_{xy}\tilde p^+_{1,j}\frac{D_x^-}{D_x^+}\tilde w^-_{0,i},-\frac{C_x^+}{C_y^+}C^+_{yy}p^+_{1,j}\tilde w^+_{0,i+1},-\frac{C_x^+}{C_y^+}C^+_{yy}p^+_{1,j}\tilde w^+_{0,i+2},0,0,C^+_{yy}p^+_{1,j+1},$$
$$C^+_{yy}p^+_{1,j+2},C^+_{xy}\tilde p^+_{1,j+\frac12},C^+_{xy}\tilde p_{1,j+\frac32}\Big),$$
$$\mathbf{c_4}^\top=\Big(0,0,D^+_{yx}w^+_{1,i+\frac12}-D^+_{xy}\tilde p^+_{1,j}\frac{D_y^+}{D_x^+}w^+_{0,i+\frac12},D^+_{yx}w^+_{1,i+\frac32}-D^+_{xy}\tilde p^+_{1,j}\frac{D_y^+}{D_x^+}w^+_{0,i+\frac32},$$
$$-D^-_{xx}\tilde w^-_{1,i-1}+D^+_{xy}\tilde p^+_{1,j}\frac{D_x^-}{D_x^+}\tilde w^-_{0,i-1},-D^-_{xx}\tilde w^-_{1,i}+D^+_{xy}\tilde p^+_{1,j}\frac{D_x^-}{D_x^+}\tilde w^-_{0,i},D^+_{xx}\tilde w^+_{1,i+1},$$
$$D^+_{xx}\tilde w^+_{1,i+2},0,0,D^+_{xy}\tilde p^+_{1,j+\frac12},D^+_{xy}\tilde p_{1,j+\frac32}\Big).$$

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

Acknowledgments

The research of Zhang was supported in part by NSFC grant 11301507. The research of Xu was supported in part by "One Hundred Talent" project of Chinese Academy of Sciences 2013 – 2015. The research of Zhao was supported in part by NSF grants DMS-1016579 and DMS-1318898 and the University of Alabama Research Stimulation Program (RSP) award.

References

[1] C. A. BALANIS, Advanced Engineering Electromagnetics, John Wiley & Sons, 1989.
[2] A. BOSSAVIT AND J. C. VERITE, *A mixed FEM-BIEM method to solve 3-D eddy current problems,* IEEE Trans. Magn., 18 (1982), pp. 431–435.
[3] W. CAI AND S. DENG, *An upwinding embedded boundary method for Maxwell's equations in media with material interfaces: 2D case,* J. Comput. Phys., 190 (2003), pp. 159–183.
[4] W. CAI, Computational Methods for Electromagnetic Phenomena: Electrostatics in Solvation, Scattering and Electron Transport, Cambridge University Press, Cambridge, 2012.
[5] M.-H. CHEN, B. COCKBURN AND F. REITICH, *High-order RKDG methods for computational electromagnetics*, J. Sci. Comput., 22 (2005), pp. 205–226.
[6] B. COCKBURN, F. LI AND C.-W. SHU, *Locally divergence-free discontinuous Galerkin methods for the Maxwell equations,* J. Comput. Phys., 194 (2004), pp. 588–610.
[7] G. COHEN, X. FERRIERES AND S. PERNET, *A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell's equations in the time domain,* J. Comput. Phys., 217 (2006), pp. 340–363.
[8] A. DITKOWSKI, K. DRIDI AND J. S. HESTHAVEN, *Convergent Cartesian grid methods for Maxwell's equations in complex geometries,* J. Comput. Phys., 170 (2001), pp. 39–80.
[9] B. DONDERICI AND F. L. TEIXEIRA, *Mixed finite-element time-domain method for Maxwell equations in doubly-dispersive media,* IEEE Trans. Microw. Theory Tech., 56 (2008), pp. 113–120.
[10] S. D. GEDNEY, C. LUO, J. A. RODEN, R. D. CRAWFORD, B. GUERNSEY, J. A. MILLER, T. KRAMER AND E. W. LUCAS, *The discontinuous Galerkin finite-element time-domain method solution of Maxwell's equations,* J. Appl. Comput. Electromagn. Soc., 24 (2009), pp. 129–142.
[11] M. GROTE, A. SCHNEEBELI AND D. SCHÖTZAU, *Interior penalty discontinuous Galerkin method for Maxwell's equations: energy norm error estimates,* J. Comput. Appl. Math., 204 (2007), pp. 375–386.
[12] J. S. HESTHAVEN, *High-order accurate methods in time-domain computational electromagnetics, A review,* Adv. Imag. Electron Phys., 127 (2003), pp. 59–123.
[13] J. S. HESTHAVEN AND T. WARBURTON, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis and Applications, Springer, New York, 2008.
[14] Y. HUANG AND J. LI, *Interior penalty discontinuous Galerkin method for Maxwell's equation in cold plasma*, J. Sci. Comput. 41 (2009), pp. 321–340.
[15] Y. HUANG, J. LI AND W. YANG, *Interior penalty DG methods for Maxwell's equations in dispersive media*, J. Comput. Phys., 230 (2011), pp. 4559–4570.
[16] L. N. T. HUYNH, N. C. NGUYEN, J. PERAIRE AND B. C. KHOO, *A high-order hybridizable discontinuous Galerkin method for elliptic interfac problems*, Int. J. Numer. Method Eng., 93 (2013), pp. 183–200.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

[17] X. Ji, W. Cai and P. W. Zhang, *High-order DGTD method for dispersive Maxwell's equations and modelling of silver nanowire coupling,* Int. J. Numer. Methods Eng., 69 (2007), pp. 308–325.
[18] D. Jiao and J. M. Jin, *Time-domain finite-element modeling of dispersive media,* IEEE Microw. Wireless Comput. Lett., 11 (2001), pp. 220–222.
[19] J. Li, *Posteriori error estimation for an interiori penalty discontinuous Galerkin method for Maxwell's equations in cold plasma*, Adv. Appl. Math. Mech., 1 (2009), pp. 107–124.
[20] J. Li, *Development of discontinuous Galerkin methods for Maxwell's equations in metamaterials and perfectly matched layers*, J. Comput. Appl. Math., 236 (2011), pp. 950–961.
[21] J. Li and Y. Huang, Time-Domain Finite Element Methods for Maxwell's Equations in Metamaterials, Springer, Berlin, 2013.
[22] J. Li and J. S. Hesthaven, *Analysis and application of the nodal discontinuous Galerkin method for wave propagation in metamaterials,* J. Comput. Phys., 258 (2014), pp. 915–930.
[23] Ping Li, Yifei Shi, Lijun Jiang and Hakan Bağci, *A hybrid time-domain discontinuous Galerkin-boundary integral method for electromagnetic scattering analysis,* IEEE Trans. Antennas Propagation, 62 (2014), pp. 2841–2846.
[24] T. Lu, P. W. Zhang and W. Cai, *Discontinuous Galerkin methods for dispersive and lossy Maxwell's equations and PML boundary conditions,* J. Comput. Phys., 200 (2004), pp. 549–580.
[25] T. Lu, P. W. Zhang and W. Cai, *Discontinuous Galerkin time domain method for GPR simulation in dispersive media,* IEEE Trans. Geosci. Remote Sensing, 43 (2005), pp. 72–80.
[26] P. Monk and G. R. Richter, *A discontinuous Galerkin method for linear symmetric hyperbolic systems in inhomogeneous media,* J. Sci. Comput., 22 (2005), pp. 443–477.
[27] E. Montseny, S. Pernet, X. Ferriéres and G. Cohen, *Dissipative terms and local time-stepping improvements in a spatial high order discontinuous Galerkin scheme for the time-domain Maxwell's equations*, J. Comput. Phys., 227 (2008), pp. 6795–6820.
[28] G. Mur and A. T. de Hoop, *A finite-element method for computing three-dimensional electromagnetic fields in inhomogeneous media,* IEEE Trans. Magn. 21 (1985), pp. 2188–2191.
[29] D. D. Nguyen and S. Zhao, *High order FDTD methods for transverse magnetic modes with dispersive interfaces,* Appl. Math. Comput., 226 (2014), pp. 699–707.
[30] D. D. Nguyen and S. Zhao, *Time-domain matched interface and boundary (MIB) modeling of Debye dispersive media with curved interfaces,* J. Comput. Phys., 278 (2014), pp. 298–325.
[31] J. C. Nédélec, *Mixed finite elements in* R^3, Numer. Meth., 35 (1980), pp. 315–341.
[32] J. C. Nédélec, *A new family of mixed finite elements in* R^3, Numer. Math., 50 (1986), pp. 57–81.
[33] N. S. Stoykov, T. A. Kuiken, M. M. Lowery and A. Taflove, *Finite-element time-domain algorithms for modeling linear Debye and Lorentz dielectric dispersions at low frequencies,* IEEE Trans. Biomed. Eng., 50 (2003), pp. 1100–1107
[34] A. Taflove and S. C. Hagness, Computational Electrodynamics: The Finite-Time Domain-Method, 3rd ed., Norwood, MA: Artech House, 2005.
[35] B. Wang, Z. Xie and Z. Zhang, *Error analysis of a discontinuous Galerkin method for Maxwell equations in dispersive media*, J. Comput. Phys., 229 (2010), pp. 8552–8563.
[36] Z. Q. Xie, C.-H. Chan and B. Zhang, *An explicit fourth order staggered finite-difference time-domain method for Maxwell's equations,* J. Comput. Appl. Math., 147 (2002), pp. 75–98.
[37] Z. Q. Xie, C.-H. Chan and B. Zhang, *An explicit fourth order orthogonal curvilinear staggered-grid FDTD method for Maxwell's equations,* J. Comput. Phys., 175 (2002), pp. 739–763.
[38] J. Xu, B. Mustapha, P. N. Ostroumov and J. Nolen, *Solving 2D2V Vlasov equation with high order spectral element method,* Commun. Comput. Phys., 8 (2010), pp. 159–184.
[39] J. Xu, P. N. Ostroumov, J. Nolen and K. J. Kim, *Scalable direct Vlasov solver with discontin-*

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press

uous Galerkin method on unstructured mesh, SIAM J. Sci. Computing, 32 (2010), pp. 3476–3494.
[40] A. YEFET AND E. TURKEL, *Fourth order compact implicit method for the Maxwell equations with discontinuous coefficients,* Appl. Numer. Math., 33 (2000), pp. 125–134.
[41] A. YEFET AND P. G. PETROPOULOS, *A staggered Fourth-order accurate explicit finite difference scheme for the time-domain Maxwell's equations,* J. Comput. Phys., 168 (2001), pp. 286–315.
[42] S. ZHAO AND G. W. WEI, *High order FDTD methods via derivative matching for Maxwell's equations with material interfaces,* J. Comput. Phys., 200 (2004), pp. 60–103.
[43] S. ZHAO, *High order matched interface and boundary methods for the Helmholtz equation in media with arbitrarily curved interfaces,* J. Comput. Phys., 229 (2010), pp. 3155–3170.
[44] S. ZHAO, *High order FDTD methods for transverse electromagnetic systems in dispersive inhomogeneous media,* Opt. Lett., 36 (2011), pp. 3245–3247.
[45] S. ZHAO, *A matched alternating direction implicit (ADI) method for solving the heat equation with interfaces,* J. Sci. Comput., 63 (2015), pp. 118–137.
[46] C. M. ZHANG AND R. J. LEVEQUE, *The immersed interface method for acoustic wave equations with discontinuous coefficients,* Wave Motion, 25 (1997), pp. 237–263.
[47] Y. C. ZHOU, S. ZHAO, M. FEIG AND G. W. WEI, *High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources,* J. Comput. Phys., 213 (2006), pp. 1–30.

https://doi.org/10.4208/aamm.2014.m811 Published online by Cambridge University Press