Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 226 (2007) 1085–1101

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A stable FDTD algorithm for non-diagonal, anisotropic dielectrics ☆

Gregory R. Werner [a,*], John R. Cary [a,b]

[a] *Center for Integrated Plasma Studies, University of Colorado, Boulder, CO 80309, United States*
[b] *Tech-X Corporation, Boulder, CO 80303, United States*

Received 9 February 2007; received in revised form 4 May 2007; accepted 7 May 2007
Available online 21 May 2007

Abstract

A stable FDTD algorithm is developed for simulating Maxwell's equations in anisotropic dielectric materials with principal axes not aligned with the grid. The algorithm is stable because the finite-difference operator that converts **D** to **E** is symmetric and positive semidefinite; for contrast, a previously developed asymmetric algorithm is shown to suffer from late-time instabilities. The presented algorithm has second-order error for continuous dielectric materials, and the error can be reduced to third-order by Richardson extrapolation. Applied to dielectrics with sharp interfaces, the algorithm has first-order error, even when averaging the dielectric within partially filled grid cells. However, averaging the dielectric permits Richardson extrapolation to obtain second-order error.

Keywords: Dielectric; Tensor dielectric; Anisotropic dielectric; Electromagnetic; FDTD; Simulation

1. Introduction

Simulating electromagnetic waves in dielectric materials is important for many applications that control electromagnetic energy, such as photonic crystals and dielectric accelerator structures. Simulating materials with tensor dielectric constants is important because many outstanding dielectrics (such as sapphire, which has very low losses at microwave frequencies) are anisotropic. Finite-difference time-domain (FDTD) algorithms allow large, efficient simulations with the flexibility to simulate complexities such as dispersive materials or charged particles interacting with electromagnetic waves.

We will develop and demonstrate a stable FDTD algorithm for simulating Maxwell's equations in non-uniform tensor dielectrics that are neither dissipative nor active. The algorithm has second-order error in continuous dielectrics, but first-order error in discontinuous dielectrics; it reduces to the standard Yee algorithm

☆ This work was supported by the US Department of Energy Grant DE-FG02-04ER41317 and Air Force Office of Scientific Research Grant FA9550-04-C-0041.

* Corresponding author. Tel.: +1 303 735 2461; fax: +1 303 492 0642.
E-mail address: Greg.Werner@colorado.edu (G.R. Werner).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.05.008

in the presence of isotropic dielectrics. (In this paper, we use "has second-order error" to describe an algorithm that, for example, computes a mode frequency f that differs from the real frequency f_0 by a quantity proportional to Δx^2, where Δx is the size of a grid cell: $f = f_0 + \mathrm{O}(\Delta x^2)$. Algorithms that have second-order error are often called second-order algorithms, though they are accurate to only first-order.)

The standard Yee FDTD algorithm [1], which has second-order error, can be easily extended to handle isotropic dielectrics. Extending that algorithm to materials with general dielectric tensors is non-trivial: algorithms that appear to be physically reasonable may lead to instabilities that appear at late times in the simulation – instabilities with growth rates unchanged by reducing the time step. The problem with tensor dielectrics occurs when the dielectric varies in space, $\underline{\underline{\varepsilon}} = \underline{\underline{\varepsilon}}(x, y, z)$, because the Yee algorithm locates the different components of the electric field at different places within a single grid cell (see Fig. 1). A tensor dielectric introduces an interaction between, e.g., x and y components of the electric field $\mathbf{E}$ and displacement $\mathbf{D}: D_y = \sum_\mu \varepsilon_{y\mu} E_\mu$. This operation, complicated by the differing locations of, e.g., E_x and D_y, can lead to instability.

This kind of instability, resulting from an asymmetry in the field update algorithm, usually grows slowly enough (though we are not aware of any guarantee of slowness) that simulations may still be useful, despite being unstable in this sense; in some cases the instability grows too slowly to be noticed at all. In particular, simulations with significant dissipation, especially scattering simulations with absorbing boundary conditions, are insensitive to this instability; indeed, several algorithms have been developed and tested in scattering simulations without noticing the instability. Even without dissipation (as in resonant cavity simulations), useful information can sometimes be extracted before the instability dominates; however, a stable algorithm takes away any worry that an instability might diminish the accuracy of a simulation.

Ref. [2] proposed an algorithm for tensor dielectrics, and determined its accuracy (second-order error) and stability in time (i.e., the maximum time step, such that any simulation with a greater time step will be unstable). However, we have found that their algorithm can lead to late-time instabilities in a non-uniform dielectric. The same algorithm was proposed earlier in [3] (which allows the dielectric to have finite conductivity) and tested on scattering simulations. Another method [4] uses an effective tensor dielectric to improve the accuracy of isotropic dielectrics with sharp interfaces. When tested with scattering simulations, where late-time instabilities would not appear, this method had smaller error than the basic Yee scheme, but still yielded first-order error. However, Ref. [5] states that this method was numerically unstable for photonic crystal band simulations. Ref. [5] describes a method for finding an effective tensor dielectric to compensate for grid cells partially filled with an isotropic dielectric; the algorithm presented in [2] is then applied to the effective dielectric. They state that they obtained second-order error when computing the lowest-frequency band of a photonic crystal with a discontinuous (isotropic) dielectric. Again, our work indicates that this algorithm is generally late-time unstable when used with effective dielectrics that have non-diagonal terms, and the effective dielectric used in [5] will have non-diagonal terms for any isotropic dielectric object with curved surfaces.

Ref. [6] describes a different FDTD scheme (based on transmission-line matrices, it does not reduce to the Yee scheme in case of isotropic dielectric) in which all field components are at the same point; however, only results from 1D scattering problems are presented. Ref. [7] presents a quasi-2D algorithm for simulating

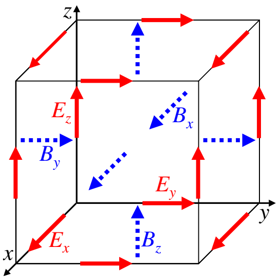

Fig. 1. Field components in one grid cell of the Yee mesh.

modes in a waveguide of uniform tensor dielectric; the algorithm is similar to that used in Refs. [2,3] (which is stable when applied to uniform dielectric). Ref. [8] treats scalar dielectric but tensor permeability using a Yee mesh with some extra components; it collocates field components in a manner resembling that of Refs. [2,3], but it is applied to lossy ferrites, where dissipation would counteract slowly growing instabilities. Ref. [9] describes simulations with a different sort of tensor dielectric, describing gyrotropic material (the dielectric tensor is complex and Hermitian, and frequency-dependent). They avoid the problem of component collocation by considering 1D scattering problems, where the non-diagonal anisotropy connects only field components located at the same point.

Other articles have offered methods for simulating isotropic dielectrics with sharp interfaces: for example [10–12] describe methods that use effective diagonal dielectrics, while [13] uses an effective tensor dielectric in 2D simulations. There are also methods that, rather than using an effective dielectric in each grid cell, alter the shapes of the cells to coincide with the dielectric shape, such as [14], which has interesting parallels to this paper, running into an accuracy versus stability trade-off, thwarted by the need to collocate different field components. While some of these methods reduce the error, they do not seem to be able to reduce the order of error to second-order.

In this paper, we develop a stable algorithm for anisotropic dielectrics in three dimensions. We start from the Yee algorithm and show four basic ways to collocate the field components so that the dielectric tensor can be applied: $E_\mu = \sum_\nu \varepsilon^{-1}_{\mu\nu} D_\nu$. Of these four ways, two yield symmetric, positive semidefinite operators and so guarantee stability (with small enough time step). We then show numerical results. We find late-time instability for the asymmetric update operators. Over the same period of time, as expected, symmetric updates remain stable. Our convergence studies for the symmetric algorithm show that it has second-order error with continuous dielectrics, and we can obtain mode frequencies with third-order error using Richardson extrapolation. Mode frequencies for discontinuous dielectrics have first-order error, even when using an effective dielectric method within each grid cell as in [5]. However, with the effective dielectric, frequencies can be obtained by Richardson extrapolation with second-order error.

The next section reviews the Yee algorithm and its stability. Section 3.1 discusses the issues of field collocation, and in Section 3.2 we determine the symmetry (hence stability) of the possible update operators. Section 4 contains our numerical results for continuous dielectrics, while Section 5 illustrates results for discontinuous dielectrics.

2. The Yee algorithm

The Yee algorithm for FDTD electromagnetics [1] integrates the dynamic Maxwell equations:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \tag{1}$$

$$\frac{\partial \mathbf{E}}{\partial t} = \nabla \times \mathbf{B}, \tag{2}$$

using finite-difference approximations. The Yee algorithm has the great advantages of simplicity, preserving important physical principles (such as $\nabla \cdot \nabla \times \mathbf{B} = 0$), and accuracy up to $\mathrm{O}(\Delta x^2)$ error, where Δx is the grid cell size. The Yee algorithm achieves second-order error by locating field components at different positions, as shown in Fig. 1, so that all finite differences in Maxwell's equations are centered (cf. [15]). Electric field component E_μ is placed at the μ-edge center, while magnetic field component B_μ is placed at the μ-face center. No two field components are known at exactly the same point.

We will now demonstrate the stability of the familiar Yee algorithm, introducing notation and concepts needed later for the dielectric algorithms. We write the discretized Maxwell's equations in matrix/operator form, with each field being a single column vector with components labeled by indices i, j, k (integers describing spatial location on the grid), and μ (x, y, or z, describing the 3D vector component): field values on the Yee mesh are

$$E_{ijk\mu} = E_\mu((i + 0.5\delta_{x\mu})\Delta x, (j + 0.5\delta_{y\mu})\Delta y, (k + 0.5\delta_{z\mu})\Delta z), \tag{3}$$

$$B_{ijk\mu} = B_\mu((i + 0.5[1 - \delta_{x\mu}])\Delta x, (j + 0.5[1 - \delta_{y\mu}])\Delta y, (k + 0.5[1 - \delta_{z\mu}])\Delta z), \tag{4}$$

where the Kronecker deltas ensure that the μ component of the electric field is placed at the μ-edge center, and the μ component of the magnetic field is at the μ-face center, as in the Yee mesh (Fig. 1).

At the heart of the algorithm is the difference operator. The forward/backward (+/−) difference operator in the x-direction is denoted by $\partial_x^\pm$; it can be represented as a matrix with rows $(ijk\mu)$ and columns $(i'j'k'\mu')$:

$$\partial^\pm_{x,(ijk\mu)(i'j'k'\mu')} = \mp\frac{\delta_{(ijk\mu)(i'j'k'\mu')} - \delta_{(ijk\mu)(i'\mp1,j'k'\mu')}}{\Delta x}, \tag{5}$$

where $\delta_{(ijk\mu)(i'j'k'\mu')} = \delta_{ii'}\delta_{jj'}\delta_{kk'}\delta_{\mu\mu'}$ is one if $i = i'$, $j = j'$, $k = k'$, and $\mu = \mu'$, and zero otherwise. ∂_x^+ is the forward difference in the x direction; for example, we could approximate $\partial E_\mu/\partial x$ at grid cell (i,j,k) by forward differencing thus:

$$(\partial_x^+ E)_{ijk\mu} = \sum_{i'j'k'\mu'} \partial^+_{x,(ijk\mu)(i'j'k'\mu')} E_{i'j'k'\mu'} = \frac{1}{\Delta x}[-E_{ijk\mu} + E_{i+1,jk\mu}], \tag{6}$$

or by backward differencing:

$$(\partial_x^- E)_{ijk\mu} = \sum_{i'j'k'\mu'} \partial^-_{x,(ijk\mu)(i'j'k'\mu')} E_{i'j'k'\mu'} = \frac{1}{\Delta x}[E_{ijk\mu} - E_{i-1,jk\mu}]. \tag{7}$$

Difference operators in the y and z directions are defined analogously, shifting the ∓1 from i' to j' and k', respectively. The y in ∂_y^+ is not a tensor component (∂_x^+, ∂_y^+, and ∂_z^+ are not the components of some vector operator); technically, ∂_x^+ is a matrix $(\partial_x^+)_{ijk\mu,i'j'k'\mu'}$, although, since it is proportional to $\delta_{\mu,\mu'}$, we often drop the μ subscripts. (We often drop the other subscripts for simplicity.)

From the difference operators we construct curl operators. Because the μth component of the curl of two vectors, v_μ and w_μ, can be written as $(v \times w)_\mu = \sum_{\nu,\kappa}\epsilon_{\mu\nu\kappa}v_\nu w_\kappa$, where ϵ is the completely antisymmetric tensor, the forward and backward curl operators can be written as

$$[\nabla^\pm \times]_{(ijk\mu)(i'j'k'\mu')} = \sum_\nu \epsilon_{\mu\nu\mu'}\partial^\pm_{\nu,(ijk)(i'j'k')}. \tag{8}$$

Applied to a vector field $E_{ijk\mu}$

$$([\nabla^\pm\times]E)_{ijk\mu} = \sum_{i'j'k'\mu'\nu} \epsilon_{\mu\nu\mu'}\partial^\pm_{\nu,(ijk)(i'j'k')}E_{i'j'k'\mu'}. \tag{9}$$

With **E** and **B** on the Yee mesh (Fig. 1), the dynamic Maxwell equations are then (in the absence of currents, and with convenient units):

$$\frac{\partial B}{\partial t} = -[\nabla^+\times]E \qquad \frac{\partial E}{\partial t} = [\nabla^-\times]B. \tag{10}$$

These equations describe the spatial discretization of the Yee algorithm. Combining them, we have

$$\frac{\partial^2 B}{\partial t^2} = -[\nabla^+\times][\nabla^-\times]B. \tag{11}$$

Here we can see the connection between the spatially discretized operator $[\nabla^+\times][\nabla^-\times]$ and the mode frequencies. Any eigenmode B behaves like a harmonic oscillator with ω^2 equal to its spatial eigenvalue. If the operator $[\nabla^+\times][\nabla^-\times]$ has a complete basis of eigenmodes with only real, non-negative eigenvalues, then ω will always be real, and no eigenmode will decay or grow with time (as noted in [16]).

As we are about to show, the operator $[\nabla^+\times][\nabla^-\times]$ is symmetric and positive semidefinite, so it is diagonalizable, and all its eigenvalues are non-negative; therefore, any solution to Eq. (11) is the sum of eigenmodes with real frequencies. First, we find the transpose of the difference operator as a matrix (suppressing the 3D vector component indices):

$$\begin{aligned}(\partial_x^\pm)^{\mathrm{T}}_{(ijk)(i'j'k')} &= \partial^\pm_{x,(i'j'k')(ijk)} = \mp\frac{1}{\Delta x}[\delta_{(i'j'k')(ijk)} - \delta_{(i'j'k')(i\mp1,jk)}] = \mp\frac{1}{\Delta x}\left[\delta_{(ijk)(i'j'k')} - \delta_{(i'\pm1,j'k')(ijk)}\right]\\ &= -\partial^\mp_{x,(ijk)(i'j'k')}.\end{aligned} \tag{12}$$

The difference operator itself is not symmetric: $(\partial_x^+)^{\mathrm{T}} = -\partial_x^- \neq \partial_x^+$. Using this result, we find the transpose of the curl operator:

$$\begin{aligned}\left([\nabla^\pm \times]_{(ijk\mu)(i'j'k'\mu')}\right)^{\mathrm{T}} &= [\nabla^\pm \times]_{(i'j'k'\mu')(ijk\mu)} = \sum_\nu \epsilon_{\mu'\nu\mu}\left(\partial^\pm_{\nu,(i'j'k')(ijk)}\right) = \sum_\nu (-\epsilon_{\mu\nu\mu'})\left(-\partial^\mp_{\nu,(ijk)(i'j'k')}\right) \\ &= [\nabla^\mp \times]_{(ijk\mu)(i'j'k'\mu')}. \end{aligned} \tag{13}$$

That is, $[\nabla^- \times]^{\mathrm{T}} = [\nabla^+ \times]$; therefore

$$[\nabla^+ \times][\nabla^- \times] = [\nabla^- \times]^{\mathrm{T}}[\nabla^- \times] \tag{14}$$

is a symmetric, positive semidefinite operator (any matrix $M = A^{\mathrm{T}}A$ is symmetric, and also positive semidefinite, because for any eigenvector v with eigenvalue λ, $\lambda\|v\|^2 = v^{\mathrm{T}}Mv = v^{\mathrm{T}}A^{\mathrm{T}}Av = \|Av\|^2 \geqslant 0$). Eigenmodes of the standard Yee algorithm therefore have real frequencies.

Just as the spatial discretization could have led to instability (if it had supported modes with complex frequencies), temporal discretization can also lead to instability. In this paper we will not be very concerned with temporal discretization, but we mention it for completeness. In the Yee scheme, the electric and magnetic fields are leap-frogged. This method is simple, efficient, and has second-order error (in the time step, Δt). For stability, the time step must be smaller than the period of the highest-frequency mode; more precisely, for the leap-frog advance the stability criterion is (cf. [15])

$$\omega_{\max}\Delta t < 2, \tag{15}$$

where $\omega_{\max}$ is the maximum frequency, or the square root of the maximum eigenvalue of the operator on the right side of Eq. (11). This leads to the well-known Courant–Friedrichs–Lewy condition for stable integration of Maxwell's equations using the Yee method. Algorithms that are stable only for small enough time steps are called "conditionally stable". Transgressing the CFL condition, which is imposed by temporal discretization, results in a catastrophic instability that manifests itself after only several time steps. In contrast, growing modes due to spatial discretization may not become apparent for tens of thousands, even hundreds of thousands of time steps, and reducing the time step does not mitigate the instability (solutions will grow at the rate determined by their complex frequencies, regardless of the time step). For the rest of this paper, we will consider stability with respect to only spatial discretization. Stability for temporal discretization can be achieved by determining $\omega_{\max}$ and setting the time step accordingly; in practice, we use $\Delta t = 0.99[c^2(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2)]^{-1/2}$.

3. Modification of the Yee algorithm for tensor dielectrics

In the presence of ideal dielectric materials (that are neither dissipative nor active), the electric displacement **D** differs from **E**. The equations to solve are Faraday's Law (Eq. (1)) and

$$\frac{\partial \mathbf{D}}{\partial t} = \nabla \times \mathbf{B}, \tag{16}$$

$$\mathbf{E} = \underline{\underline{\xi}}\mathbf{D}, \tag{17}$$

where $\underline{\underline{\xi}} = \underline{\underline{\varepsilon}}^{-1}$. If **B** and **E** are located as usual on the Yee mesh (see Fig. 1), and **D** is located at the same places as **E**, then we already know how to perform the FDTD update of Eq. (1) and the first equation above, according to the Yee algorithm, with second-order error. In the next section we will concentrate on the equation converting between **E** and **D** for a smoothly varying dielectric. Then we will discuss the stability of the resulting algorithm.

3.1. Collocation of field components for conversion between *E* and *D*

To convert from **D** to **E** in continuous space, one simply multiplies by the inverse dielectric tensor: $E_\nu = \sum_\mu \xi_{\nu\mu} D_\mu$, where E_ν, $\xi_{\nu\mu}$, and D_μ are all evaluated at the same point. In our discretized approximation, however, we do not know, e.g., D_x and D_y at the same point. Simply calculating $E_x = \sum_\mu \xi_{x\mu} D_\mu$, taking D_μ

from the nearest locations, yields E_x with an error proportional to the grid cell length Δx, even if the D_μ are exact. To achieve second-order error, one must interpolate the D_μ to the location of E_x: for example, if D_x is known at the centers of the x-edges in a cell:

$$D_x(0,y,z) = \frac{1}{2}[D_x(-\Delta x/2,y,z) + D_x(\Delta x/2,y,z)] + \mathrm{O}(\Delta x^2) \tag{18}$$

is an interpolation with second-order error – as long as D_x has a continuous derivative in the x direction. Such an interpolation can accurately collocate components within a smoothly varying dielectric, but not around a sharp dielectric interface, where **D** is discontinuous; we will address sharp interfaces in Section 5.

We now introduce interpolation operators. The interpolation operator for the x-direction is

$$I^{\pm}_{x,(ijk\mu)(i'j'k'\mu')} = \frac{1}{2}[\delta_{(ijk\mu)(i'j'k'\mu')} + \delta_{(ijk\mu)(i'\mp 1,j'k'\mu')}]. \tag{19}$$

$I^{\pm}_y$ and $I^{\pm}_z$ are analogously defined. For example, the y component of the electric field interpolated to the point half-way between (i,j,k) and $(i,j+1,k)$ would be

$$(I^+_y E)_{ijky} = \frac{1}{2}[E_{ijky} + E_{i,j+1,ky}]. \tag{20}$$

The $I^{\pm}_\mu$ all commute; for all μ and ν: $0 = [I^+_\mu, I^+_\nu] = [I^-_\mu, I^-_\nu] = [I^+_\mu, I^-_\nu]$.

With these operators, we describe the interpolation of the components of **D** to the location of E_x with second-order error:

$$D(\text{at } E_x)_{ijkx} = D_{ijkx}, \tag{21}$$

$$D(\text{at } E_x)_{ijky} = (I^-_y I^+_x D_y)_{ijk} \qquad \text{(see Fig. 2a)}, \tag{22}$$

$$D(\text{at } E_x)_{ijkz} = (I^-_z I^+_x D_z)_{ijk}. \tag{23}$$

The D_x to $D(\text{at } E_x)_x$ conversion is trivial since D_x is already at the same place as E_x.

Once we know the components of **D** at the same location, we can perform the matrix multiplication by the inverse dielectric tensor $\xi = \varepsilon^{-1}$ to get **E** at that location. Treating ξ as a matrix diagonal in the spatial indices:

$$\xi_{(ijk\mu)(i'j'k'\nu)} = \xi_{ijk\mu\nu}\delta_{(ijk)(i'j'k')}, \tag{24}$$

we can write a shorthand matrix equation that shows the vector components explicitly, but hides the spatial components, so that

$$(E_x)_y = \xi_{xy} I^+_x I^-_y D_y \tag{25}$$

describes the interpolation of D_y to the location of E_x and the subsequent multiplication by ξ_{xy} (as shown in Fig. 2a).

We now consider other ways of collocation and application of the inverse dielectric tensor. For $\mu = y$, there are four "simple" possibilities with second-order error, illustrated in Fig. 2:

$$\begin{aligned}
&\text{(a)} \quad (E_x)_y = \xi_{xy}(x\text{-edge-center}) I^+_x I^-_y D_y = \xi^{x\mathrm{e}}_{xy} I^+_x I^-_y D_y,\\
&\text{(b)} \quad (E_x)_y = I^+_x I^-_y \xi_{xy}(y\text{-edge-center}) D_y = I^+_x I^-_y \xi^{y\mathrm{e}}_{xy} D_y,\\
&\text{(c)} \quad (E_x)_y = I^+_x \xi_{xy}(\text{node}) I^-_y D_y = I^+_x \xi^{\mathrm{n}}_{xy} I^-_y D_y,\\
&\text{(d)} \quad (E_x)_y = I^-_y \xi_{xy}(z\text{-face-center}) I^+_x D_y = I^-_y \xi^{xy\mathrm{f}}_{xy} I^+_x D_y.
\end{aligned} \tag{26}$$

The same options are available for $(E_x)_z$, *mutatis mutandis*. We designate the location of ξ with superscript abbreviations: xe for x-edge-center; xyf for z-face-center; and n for node.

For $(E_x)_x$, we have two possibilities:

$$\begin{aligned}
&\text{(e)} \quad (E_x)_x = \xi_{xx}(x\text{-edge}) D_x = \xi^{x\mathrm{e}}_{xx} D_x,\\
&\text{(f)} \quad (E_x)_x = I^+_x \xi_{xy}(\text{node}) I^-_x D_x = I^+_x \xi^{\mathrm{n}}_{xy} I^-_x D_x.
\end{aligned} \tag{27}$$

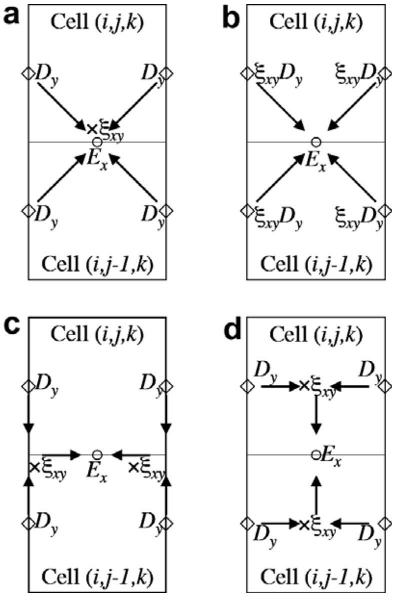

Fig. 2. The four methods of interpolating and multiplying by ξ_{xy} to get the part of E_x that comes from D_y, from Eq. (26).

Choosing option (27e) will ensure that this algorithm reduces to the Yee algorithm when the dielectric tensor is diagonal. Option (27f), when paired with option (26c), has an attractive consistency of form and puts all dielectric components at the cell node; unfortunately, this option has poor dispersion (the phase velocity of light goes to zero at short wavelengths for certain directions and polarizations, so that a low-frequency excitation may excite an unphysical, short-wavelength mode).

We can write $E_\mu = \sum_\nu \Xi_{\mu\nu} D_\nu$ or, more carefully:

$$E_{ijk\mu} = \sum_{i'j'k'\mu'} \Xi_{(ijk\mu)(i'j'k'\mu')} D_{i'j'k'\mu'}, \tag{28}$$

as the method for updating the electric field from the displacement. This "inverse dielectric tensor" operation on the grid is not as simple as multiplying by a simple 3 × 3 matrix in each grid cell:

$$\Xi_{(ijk\mu)(i'j'k'\nu)} \neq \xi_{ijk\mu\nu}\delta_{(ijk)(i'j'k')}. \tag{29}$$

To achieve second-order error, as we've discussed, we need a centered interpolation of **D**. For example, if we use scheme (26a) the matrix Ξ will have elements like

$$\Xi_{(ijkx)(i''j''k''y)} = \xi^{xe}_{ijkxy} \sum_{i'j'k'} I^{+}_{x,(ijky)(i'j'k'y)} I^{-}_{y,(i'j'k'y)(i''j''k''y)}, \tag{30}$$

where ξ^{xe}_{ijkxy} is the xy component of the inverse dielectric tensor at the location of E_{ijkx}. While the schemes in this section have second-order error, they are not all stable; we will discuss stability in the next section.

3.2. Ensuring diagonalizability and non-negative eigenvalues for stability

Spatial discretization can lead to instability if the resulting eigenmodes can have complex frequencies. In continuous space, solutions to Maxwell's sourceless equations neither grow nor decay in time – field energy is conserved (as long as dielectric media are neither dissipative nor active). In discrete space, however, there may be growing or decaying modes if the discretized update operator has complex eigenvalues.

The standard Yee update (see Section 2) is stable because its discretization of the ∇×∇× operator is symmetric and positive semidefinite – it is diagonalizable, and its eigenvalues are real and non-negative; mode frequencies, square roots of those eigenvalues, are therefore real. To show that our algorithm is stable with tensor dielectrics, we need to show that the discretization of the $\nabla \times \underline{\underline{\xi}} \nabla \times$ operator is symmetric and positive semidefinite.

In the presence of tensor dielectrics, the time-advance of **E** becomes (in continuous space)

$$\partial \mathbf{E}/\partial t = \underline{\underline{\xi}} \nabla \times \mathbf{B}, \tag{31}$$

where $\underline{\underline{\xi}}$ is the inverse dielectric tensor. The discretization of this equation leads to the equations

$$\frac{\partial B}{\partial t} = -[\nabla^+ \times] E \qquad \frac{\partial E}{\partial t} = \Xi [\nabla^- \times] B \tag{32}$$

or

$$\omega^2 B = [\nabla^+ \times] \Xi [\nabla^- \times] B, \tag{33}$$

where Ξ is the discretized "inverse dielectric tensor" from Section 3.1.

If $\Xi^{\mathrm{T}} = \Xi$, then the operator $[\nabla^+ \times] \Xi [\nabla^- \times]$ is symmetric; if, in addition, Ξ is positive semidefinite, then the operator is positive semidefinite as well. (A symmetric, positive semidefinite matrix has a symmetric square root; if Ξ is symmetric and positive semidefinite, there exists a matrix A such that $A^2 = A^{\mathrm{T}} A = \Xi$. Therefore $[\nabla^+ \times] \Xi [\nabla^- \times] = (A[\nabla^- \times])^{\mathrm{T}} A [\nabla^- \times]$ is symmetric and positive semidefinite too. We showed in Section 2 that any matrix that can be written $B^{\mathrm{T}} B$ for some operator B is symmetric and positive semidefinite. To see that a symmetric, positive semidefinite matrix Ξ has a square root, diagonalize it: $\Xi = R^{\mathrm{T}} D R$ for some orthogonal matrix R and non-negative, diagonal matrix D. The square root of Ξ is $R^{\mathrm{T}} \sqrt{D} R$.)

We can now examine the algorithms from Section 3.1 to see whether they are symmetric and positive semidefinite. The algorithm combining (26a) and (27e) can be written thus:

$$\Xi_{\mu\nu} = \delta_{\mu\nu} \xi_{\mu\mu} + (1 - \delta_{\mu\nu}) \xi_{\mu\nu} I_\mu^+ I_\nu^-. \tag{34}$$

Its transpose, since $(I_\mu^+)^{\mathrm{T}} = I_\mu^-$, is

$$(\Xi)_{\mu\nu}^{\mathrm{T}} = \delta_{\nu\mu} \xi_{\nu\nu} + (1 - \delta_{\nu\mu})(I_\mu^-)^{\mathrm{T}} (I_\nu^+)^{\mathrm{T}} \xi_{\nu\mu} = \delta_{\mu\nu} \xi_{\mu\mu} + (1 - \delta_{\mu\nu}) I_\mu^+ I_\nu^- \xi_{\nu\mu}. \tag{35}$$

For an arbitrary tensor dielectric varying in space, Ξ is not symmetric with schemes (26a) and (27e) even when the dielectric tensor is symmetric at every point in space; Ξ is not symmetric because (1) the interpolation ruins the symmetry, and (2) $\xi_{ijk\mu\nu}$ need not equal $\xi_{ijk\nu\mu}$ because they are located at different places. If ξ is uniform, however, then Ξ is symmetric.

Of the four ways of finding $(E_x)_y$, (26c) and (26d) are symmetric and positive semidefinite when Ξ is symmetric and positive semidefinite, while (26a) and (26b) are transposes of each other. Both ways of finding $(E_x)_x$ lead to symmetric, positive semidefinite algorithms. Although a linear combination of these different algorithms for conversion between **D** and **E** would still have second-order error, even a symmetric linear combination might not be positive semidefinite. Ruling out (27f) because of its poor dispersion, we are most interested in the combination of (27e) along with either (26c) or (26d), both of which produce positive semidefinite Ξ. We will demonstrate the algorithms (27e) and (26c) in the next section.

4. Numerical results for continuously varying dielectrics

In the previous section, we found two symmetric and positive semidefinite algorithms for finite difference electromagnetics with tensor dielectrics; those algorithms are guaranteed to be stable (as long as the time step is small enough). We could not determine analytically whether the other two (asymmetric) algorithms were diagonalizable and positive semidefinite, but we will demonstrate experimentally that one of those algorithms does indeed lead to instability (no matter how small the time step). On the other hand, the same simulation experiences no instability when using a symmetric, positive semidefinite algorithm; we will demonstrate the second-order error of this stable algorithm. Finally, we will show how Richardson extrapolation can reduce the error by another order, using the frequencies extracted from two simulations with different grid resolutions.

Both algorithms, stable and unstable, were implemented within the VORPAL [17] simulation framework. We then calculated band frequencies of a cubic lattice (with lattice constant a) of spheres, with dielectric varying smoothly from sapphire to vacuum, for a given Bloch wave-vector (for an introduction to photonic crystals, see, e.g. [18]). The dielectric tensor at the center of the spheres was set to

$$\underline{\underline{\varepsilon}} = \begin{pmatrix} 10.225 & -0.825 & -0.55\sqrt{3/2} \\ -0.825 & 10.225 & 0.55\sqrt{3/2} \\ -0.55\sqrt{3/2} & 0.55\sqrt{3/2} & 9.95 \end{pmatrix}\varepsilon_0, \tag{36}$$

which is the dielectric constant for sapphire, with $\varepsilon_{\|} = 11.6\varepsilon_0$ along the y-direction and $\varepsilon_{\perp} = 9.4\varepsilon_0$ perpendicular to y, subsequently rotated 30° about the x-axis, and then 45° about the z-axis. We then made a smooth transition between sapphire and vacuum by interpolating each element of the dielectric tensor $\underline{\underline{\varepsilon}}$ from sapphire at radius (squared) $r^2 = (0.3a)^2$ to vacuum at $r^2 = (0.45a)^2$. We used a cubic interpolation, demanding that the derivatives vanish at the end points. (A linear interpolation, with discontinuous first derivatives, does not spoil the order of accuracy of the algorithm.)

In these simulations, we analyzed a single unit cell of the lattice, with lattice constant $a = 1$ m, imposing phase-shifted periodic boundary conditions to find mode frequencies corresponding to the Bloch wave-vector $\mathbf{k} = (2\pi/a) \times (0.1, 0.2, 0.3)$. We excited the cavity with a current pulse $t\,\mathrm{e}^{-t^2/2\sigma^2}$ (meant to excite a broad range of low frequencies) at a point, with $\sigma = a/c$ where c is the speed of light in vacuum. Then we tracked a field component at every time step (which was 0.99 times the Courant–Friedrichs–Lewy maximum time step), and then Fourier-transformed the time-series to find the mode frequencies, using a vocoder-based method to extract precise values (see Appendix A).

To illustrate that instability can occur for asymmetric update operators, we implemented the schemes (26a) and (27e) described in Section 3.1 (also described in Refs. [2,3,5]):

$$\Delta E_\mu = \Delta t\left[\xi^{\mu\mathrm{e}}_{\mu\mu}([\nabla^-\times]B)_\mu + \sum_{\nu\neq\mu}\xi^{\mu\mathrm{e}}_{\mu\nu}I^+_\mu I^-_\nu([\nabla^-\times]B)_\nu\right]. \tag{37}$$

The magnetic field is updated in the usual way,

$$\Delta\mathbf{B} = -\Delta t[\nabla^+\times]\mathbf{E}. \tag{38}$$

Fig. 3 shows the resulting time-behavior using an $8\times8\times8$ grid for the lattice unit cell; the instability is not apparent for tens of thousands of time steps (here 10,000 time steps is nearly 200 periods of the lowest mode) but subsequently overwhelms the simulation. The instability grows as $\mathrm{e}^{\gamma t}$, where $\gamma/\omega_0 \approx 2\times10^{-3}$ and ω_0 is the frequency of the lowest mode ($1.61c/a$).

The instability is not present with the symmetric, positive semidefinite algorithm obtained by combining (26c) and (27e) in Section 3.1:

$$\Delta E_\mu = \Delta t\left[\xi^{\mu\mathrm{e}}_{\mu\mu}([\nabla^-\times]B)_\mu + \sum_{\nu\neq\mu}I^+_\mu\xi^{\mathrm{n}}_{\mu\nu}I^-_\nu([\nabla^-\times]B)_\nu\right]. \tag{39}$$

With this algorithm, off-diagonal elements of ξ are located at the nodes of each grid cell (enforcing $\xi_{\mu\nu} = \xi_{\nu\mu}$), while the diagonal elements $\xi_{\mu\mu}$ are located with E_μ. This algorithm shows no instability, even for simulation times much longer than those of Fig. 3.

Able to run simulations stably for long times, we tracked a field component at every time step, and extracted the frequencies present. Fig. 4 shows the relative error in frequencies for the lowest 8 modes, compared against our best-value frequencies, which we got by Richardson-extrapolating from $N = 64$ and $N = 128$ (grid cells per lattice constant) assuming second-order error. The results show second-order error in $\Delta x \propto 1/N$. Even more convincing, Richardson extrapolation (assuming second-order error) from successive simulations, $N = 8$ and $N = 16$, then $N = 16$ and $N = 32$, and so on, yields frequencies with roughly third-order error (Fig. 5).

The two lowest bands usually yield lower error than the other bands, because they have relatively long wavelengths that render them insensitive to small-scale variations in the dielectric. Instead they behave as if in a uniform medium, somewhere between vacuum and dielectric; a band diagram reflects this, showing these two lowest modes (one for each polarization) in the region of linear dispersion (plane-wave propagation at constant speed, as in a uniform medium).

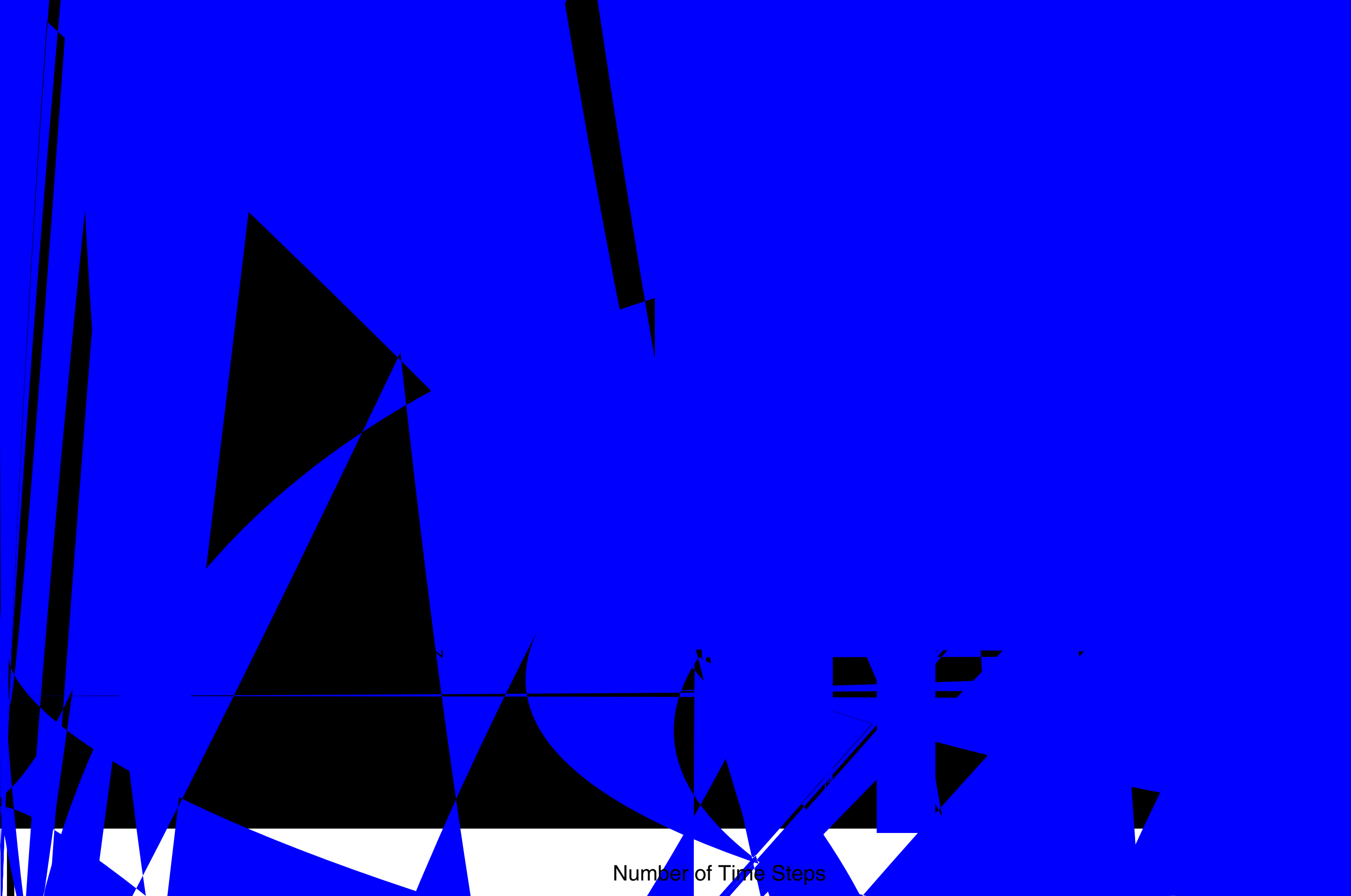

Fig. 3. The electric field E_z versus time for a simulation with an asymmetric update algorithm; the reduced vertical scale in the lower figure shows that the instability manifests itself only after nearly 30,000 time steps.

5. Nu[illegible]erical results [illegible]

[illegible] ency-domain [illegible] tonic bands code [19] [illegible] chie [illegible] rd [illegible] method [illegible] m a discontinuous d[illegible]ric ten[illegible] di[illegible] averag[illegible] ethod to fi[illegible] he effec[illegible] dielectr[illegible] each [illegible] dielectric [illegible] or for th[illegible] e-cell volume cen[illegible] on of [illegible] gonal el[illegible] nts f[illegible] ive diel[illegible] ntered [illegible] ive dielectric of the ce[illegible] volume centered [illegible] n of E_μ. [illegible] rse) di[illegible] tensor, we tested our [illegible] lattice [illegible] dielectric tensor given in Eq [illegible] c [illegible] und th[illegible] by [illegible] ma[illegible] the n[illegible] photonic [illegible] ichards[illegible] ola[illegible] er [illegible] these b[illegible]

[illegible] frequencies, co[illegible] he r[illegible] of grid [illegible] rs seco[illegible] olution, but [illegible] nges to first-o[illegible] first two [illegible] y [illegible] larger than the lattic[illegible] lly unifo[illegible] ive medium

The overall error is first order, although the error in updating field components in grid cells at the dielectric interface is zeroth order (since **D** is discontinuous at the interface), because there are many fewer cells at the interface than in the entire simulation volume. Gustafsson's work [20] suggests that in a simulation of N^d cells

Table 1
Our best estimate for the frequencies of the first eight modes, obtained using the MIT photonic bands code and Richardson extrapolation from grids of size 64^3 and 128^3

Mode	Frequency f (c/a)
1	0.261373
2	0.269798
3	0.389460
4	0.411564
5	0.422541
6	0.498700
7	0.519614
8	0.537272

Frequencies are given in units of c/a, where c is the speed of light, and a the lattice constant.

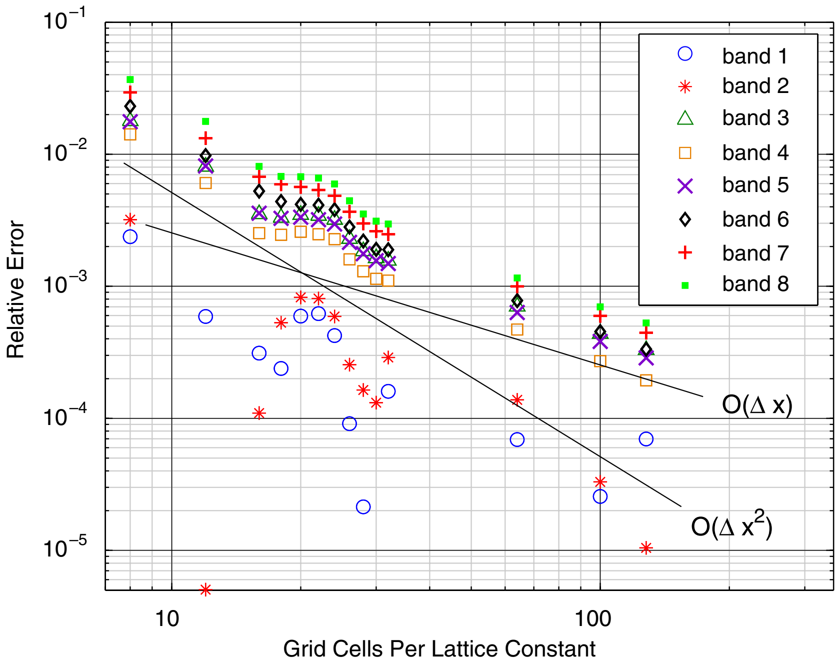

Fig. 6. Relative error in mode frequencies of the lowest eight modes versus number of grid cells divisions per length a.

(where d is the dimension), the error can be one order worse on $O(N^{d-1})$ cells, or two orders worse on $O(N^{d-2})$ cells without changing the global order of error.

The observed first-order error seems to stem from the interpolation of a discontinuous field across sharp dielectric interfaces for the following reasons. Continuous dielectric simulations yield second-order error; modes in 2D simulations (of dielectric rods) with the electric field perpendicular to the simulation plane also yield second-order error, because the electric field is continuous (2D modes with the electric field in the plane, however, yield first-order error). In addition, isotropic dielectric cubes aligned with the grid axes also yield second-order error when the dielectric is averaged as above (since, except at corners of the dielectric object, no interpolation of field components is needed because the effective dielectric is diagonal).

The apparent second-order error shown in Fig. 6 for coarse resolutions might be explained by the discontinuities in the fields at dielectric interfaces being comparable to or less than the change in the fields over a single grid cell; in such cases, the error in interpolating a discontinuous function at the dielectric interface becomes relatively less important than the errors introduced by the finite-difference approximation to the spatial derivatives over the entire volume.

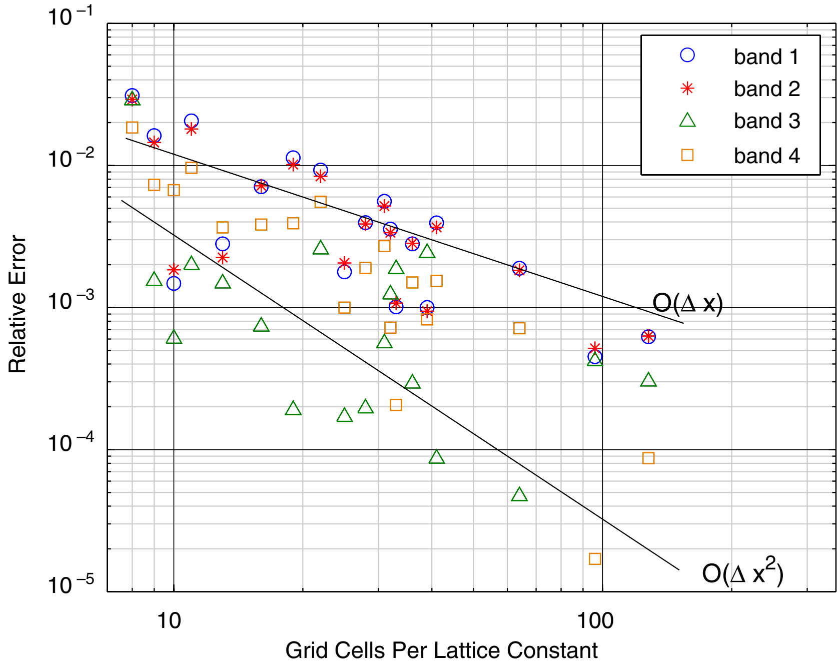

Fig. 7. Relative error in mode frequencies of the lowest four modes versus number of grid cell divisions per length *a*, using stair-stepped dielectric.

Although using an averaged dielectric in partially filled grid cells did not reduce the error of an individual simulation to second order, it did allow Richardson extrapolation from two simulations, one at half the resolution of the other, to obtain frequencies with second-order error. In contrast, using a stair-stepped dielectric resulted in frequencies that fluctuated wildly with resolution (as shown in Fig. 7), preventing extrapolation to higher order. To compute average dielectric quantities within a single grid-cell volume, we found it necessary (for Richardson extrapolation) to approximate the dielectric interface in each cell-volume with several planar triangles and to calculate the fraction of the cell volume inside and outside the approximate surface. (Richardson extrapolation did not yield higher order error when we simply averaged dielectric quantities at roughly 300 points within each grid cell.)

6. Conclusion

We have developed an algorithm that stably advances electromagnetic fields defined on a Yee mesh by ensuring that the inverse-epsilon matrix for the spatially differenced system is symmetric, even when the principal axes of the dielectric tensor are not aligned with the grid. For contrast we showed that other options for obtaining **E** from **D**, while having the same order of accuracy, are subject to late-time instabilities. The algorithm therefore allows materials such as sapphire and quartz to be stably simulated in any orientation. For continuously varying dielectrics we demonstrated second-order error of the algorithm (in the grid cell size Δx) by computing band frequencies of a photonic crystal. Using an effective dielectric in each grid cell, dielectrics with sharp interfaces can be simulated with first-order error, and the results can be Richardson-extrapolated to yield frequencies with second-order error.

Our results seem to be at odds with those of [5], in which the authors show apparent second-order error by using the algorithms (26a) and (27e), along with the effective dielectric method derived in [19]. In contrast, we see first-order error, even though the symmetric algorithm we demonstrated is very similar to the asymmetric algorithm of Ref. [5]. Indeed, we see this first-order error even for isotropic, though discontinuous, dielectrics (for isotropic dielectric objects with curved surfaces, the effective dielectric method of [19] leads to anisotropic tensor dielectric in partially filled grid cells). The resolution of this contradiction is unclear. It might be due to the unusually low error (observed here) in the first mode, since [5] considers only the lowest mode. It might also be related to the apparent necessity of averaging the effective dielectric of a two-cell-length region in [5] to obtain second-order error. (The results from averaging over a single grid cell did not show very convincing second-order behavior.) Perhaps the two-cell-length averaging simply moves the second- to first-order transition (which we observe) to a finer resolution by increasing the error at coarse resolutions. Further study is needed to resolve this issue definitively.

We also found that the algorithm of [5] can be unstable, even for isotropic, discontinuous dielectrics (which, as noted, lead to anisotropic dielectric in partially filled grid cells). In this case, the resolution may simply be that instability was not seen in [5] because it takes a long time to appear.

Acknowledgments

The implementation of dielectric algorithms was made easy by previous development of VORPAL, for which we acknowledge support from the Offices of FES, HEP, and NP of the Department of Energy, the SciDAC program, AFOSR, JTO, Office of the Secretary of Defense, and the SBIR programs of the Department of Energy and Department of Defense. We also acknowledge assistance from the VORPAL team: G.I. Bell, D.L. Bruhwiler, R.S. Busby, J. Carlsson, D.A. Dimitrov, A. Hakim, P. Messmer, P. Mullowney, C. Nieter, K. Paul, S.W. Sides, N.D. Sizemore, D.N. Smithe, P.H. Stoltz, R. Trines, S.A. Veitzer, D.J. Wade-Stein, W.-L. Wang, and N. Xiang.

Appendix A. Extracting mode frequencies

To extract mode frequencies and amplitudes from a time series $f(t_n)$, where $t_n = n\Delta t$, $n = 0, 1, 2, \ldots, N - 1 + M$, we can start with a Fourier transform. However, the Fourier transform is limited to a frequency accuracy of $1/T$, where T is the length of time over which the data are sampled. If it is known that $f(t)$ is the sum of several isolated sinusoids, then one can measure the frequencies with much greater precision. If the mode frequencies are separated by at least a few $1/T$, we have used the following method, based on the vocoder method (see, for instance [21]), to extract precise frequencies quickly. The phase vocoder extracts phase-shift information by comparing the Fourier transform of the timeseries with the transform of the timeseries shifted by a small time $\tau = M\Delta t$. Alternatively, one may prefer the method of [22] to extract frequencies from time series data.

To facilitate the analysis, we introduce a correspondence between lower case functions $g(t_n)$, and upper case functions $G(t_n)$. We will consider $g(t_n)$ to be defined for $n = 0, 1, \ldots, N - 1 + M$, while $G(t_n)$ is defined for all integer n, with $g(t_n) = G(t_n)$ for $0 \leqslant n < N + M$. This distinction makes it clear that the Fourier transform of g is defined for discrete frequencies, whereas the Fourier transform of G is defined for continuous frequencies.

We define a function $F(t_n)$ for all n, but equal to $f(t_n)$ for $n = 0, 1, \ldots, N - 1 + M$. Similarly, we will define a window function $H(t_n)$, which is zero for $n < 0$ and $n \geqslant N + M$, and $h(t_n) = H(t_n)$ is defined only for $0 \leqslant n < N + M$.

If $f(t)$, or $F(t)$, is the sum of J different sinusoids:

$$F(t) = \sum_{j=1}^{J} A_j \mathrm{e}^{\mathrm{i}\Omega_j t}, \tag{A.1}$$

then the (windowed) Fourier transform of $G(t) = H(t)F(t)$ is

$$\widetilde{G}(\omega) \equiv \frac{\Delta t}{2\pi} \sum_{n=-\infty}^{\infty} H(t_n)F(t_n)\,\mathrm{e}^{-\mathrm{i}\omega t_n} = \sum_j A_j \widetilde{H}(\omega - \Omega_j). \tag{A.2}$$

The windowed Fourier transform of $F(t)$ shifted by time τ is the transform of $G_\tau(t) \equiv H(t)F(t + \tau)$:

$$\widetilde{G}_\tau(\omega) = \frac{\Delta t}{2\pi} \sum_n H(t_n)F(t_n + \tau)\,\mathrm{e}^{-\mathrm{i}\omega t_n} = \frac{\Delta t}{2\pi} \sum_{n,j} H(t_n)A_j\,\mathrm{e}^{\mathrm{i}\Omega_j(t_n+\tau)}\,\mathrm{e}^{-\mathrm{i}\omega t_n} = \sum_j A_j \widetilde{H}(\omega - \Omega_j)\,\mathrm{e}^{\mathrm{i}\Omega_j \tau}. \tag{A.3}$$

Since $G(t_n)$ vanishes except for $0 \leqslant n < N$

$$\tilde{g}(\omega_m) \equiv \frac{1}{N}\sum_{n=0}^{N-1} g(t_n)\,\mathrm{e}^{-\mathrm{i}\omega_m t_n} = \frac{2\pi}{N\Delta t}\widetilde{G}(\omega_m) \tag{A.4}$$

for $\omega_m = m\frac{2\pi}{N\Delta t}$ (where m is an integer).

The vocoder method exploits the relationship between G and G_τ. If $f(t)$ is a single sinusoid, then

$$\frac{\tilde{g}_\tau(\omega_m)}{\tilde{g}(\omega_m)} = \mathrm{e}^{\mathrm{i}\Omega\tau}. \tag{A.5}$$

Therefore, one can extract Ω from the Fourier transforms of g and g_τ (at any ω_m). If $f(t)$ has more than one sinusoid, then

$$\frac{\tilde{g}_\tau(\omega_m)}{\tilde{g}(\omega_m)} = \frac{\sum_j A_j \widetilde{H}(\omega_m - \Omega_j)\,\mathrm{e}^{\mathrm{i}\Omega_j\tau}}{\sum_j A_j \widetilde{H}(\omega_m - \Omega_j)} = \frac{A_k \widetilde{H}(\omega_m - \Omega_k)\,\mathrm{e}^{\mathrm{i}\Omega_k\tau} + \sum_{j\neq k} A_j \widetilde{H}(\omega_m - \Omega_j)\,\mathrm{e}^{\mathrm{i}\Omega_j\tau}}{A_k \widetilde{H}(\omega_m - \Omega_k) + \sum_{j\neq k} A_j \widetilde{H}(\omega_m - \Omega_j)}. \tag{A.6}$$

The window function $\widetilde{H}(\omega)$ is steeply peaked at $\omega = 0$, so that when ω_m is within the same frequency bin as Ω_k, and all other Ω_j are far away, the $j \neq k$ terms can be neglected in the above, and

$$\mathrm{e}^{\mathrm{i}\Omega_k\tau} \approx \frac{\tilde{g}_\tau(\omega_m)}{\tilde{g}(\omega_m)}, \tag{A.7}$$

and (now that Ω_k has been found)

$$A_k \approx \frac{\widetilde{G}(\omega_m)}{\widetilde{H}(\omega_m - \Omega_k)}. \tag{A.8}$$

Of course, one first has to determine which ω_m are close to an Ω_k present in the signal. This is not too difficult; one simply loops through all the ω_m, calculating

$$\Omega_m = \frac{1}{\tau}\,\mathrm{ph}\,\frac{\tilde{g}_\tau(\omega_m)}{\tilde{g}(\omega_m)}, \tag{A.9}$$

for each ω_m, and keeps the mode only if Ω_m is close to ω_m. If $\tau > \Delta t$ (or $M > 1$), one may have to add an appropriate multiple of $2\pi/\tau$ to Ω_m (to make it as close to ω_m as possible). We have simply used $\tau = \Delta t$.

Thus far we have performed two Fourier transforms (though we have not pursued this, with $\tau = \Delta t$ it is possible to do only one transform [21]), and then performed a division operation for each ω_m (within the range of interest), and finally estimated the amplitudes of the modes.

Once one has obtained approximate frequencies, $\Omega_j^{(0)}$, and amplitudes, $A_j^{(0)}$, for all the modes j, one can refine the results:

$$\exp\left[\mathrm{i}\Omega_k^{(n+1)}\tau\right] = \frac{\widetilde{G}(\omega_m) - \sum_{j\neq k} A_j^{(n)} \widetilde{H}\left(\omega_m - \Omega_j^{(n)}\right)\mathrm{e}^{\mathrm{i}\Omega_j^{(n)}\tau}}{\widetilde{G}_\tau(\omega_m) - \sum_{j\neq k} A_j^{(n)} \widetilde{H}\left(\omega_m - \Omega_j^{(n)}\right)}, \tag{A.10}$$

$$A_k^{(n+1)} = \frac{\widetilde{G}(\omega_m) - \sum_{j\neq k} A_j^{(n)} \widetilde{H}\left(\omega_m - \Omega_j^{(n)}\right)}{\widetilde{H}\left(\omega_m - \Omega_k^{(n+1)}\right)}, \tag{A.11}$$

where it is understood that in the above, ω_m is the closest to $\Omega_k^{(n)}$ (and both may change with iteration).

If $\widetilde{H}(\omega)$ can be evaluated easily (probably analytically) for continuous ω, then each iteration above requires a number of operations proportional to the number of modes. This method is therefore very efficient for finding isolated modes with a very long time series. Although finding modes within a limited frequency range could be better done by filtering first, we have had success simply by keeping only those $\Omega^{(0)}$ within range and proceeding with the refinement. One can perform 10 iterations identifying 30 modes from a series with $N = 10^5$ in a few seconds; these iterations may reduce the change in frequency and amplitude from each iteration down below one part in 10^{14}. The choice of window is important; we have used a Hann window, $\sin^2(\pi n/N)$, because it is analytically tractable and has fairly good discrimination properties.

We have assumed that all frequencies are real (and in our photonic crystal band simulations, they must be real, since the algorithm is stable, and there is no dissipation). An advantage of this method is that one finds a quantity that is supposed to be $e^{i\Omega_k t}$, which must have unity magnitude; one can tell, to some extent, how much other sinusoids interfere by how far that quantity is from the origin. For cases in which there might be some dissipation, we expect this method (slightly modified) to be able to find modes with complex frequencies, but we have not tried to do this.

So far we have found this method to be remarkably simple and robust, especially when peaks are well-separated; however, it is very sensitive to mode frequency changes in time, so one must wait until after any driver has turned off before collecting the time series. During iteration, modes (or rather, their estimated frequencies) may move around; a good algorithm should note this movement and merge two frequencies if they get close together. For example, initial guesses from nearby bins ω_m and $\omega_{m'}$ may yield frequencies a few bins apart, apparently two different modes; after subtracting out other frequencies, however, it may become apparent that only one mode is present between those frequencies, and the algorithm must realize that. Also, noise may sometimes lead to identification of a non-existent mode; during iteration, say $\Omega_j^{(1)}$ may be very far from $\Omega_j^{(0)}$, indicating that there was never a mode near $\Omega_j^{(1)}$, and that frequency must be discarded. We have found that padding the initial time series with zeros (increasing its length by about a factor of two or so) somewhat reduces problems with mode misidentification.

References

[1] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propag. 14 (3) (1966) 302.

[2] S.G. García, T.M. Hung-Bao, R.G. Martín, B.G. Olmedo, On the application of finite methods in time domain to anisotropic dielectric waveguides, IEEE Trans. Microwave Theory Tech. 44 (12) (1996) 2195.

[3] J. Schneider, S. Hudson, The finite-difference time-domain method applied to anisotropic material, IEEE Trans. Antennas Propag. 41 (7) (1993) 994.

[4] J. Nadobny, D. Sullivan, W. Wlodarczyk, P. Deuflhard, P. Wust, A 3-D tensor FDTD-formulation for treatment of sloped interfaces in electrically inhomogeneous media, IEEE Trans. Antennas Propag. 51 (8) (2003) 1760.

[5] A. Farjadpour, D. Roundy, A. Rodriguez, M. Ibanescu, P. Bermel, J.D. Joannopoulos, S.G. Johnson, Improving accuracy by subpixel smoothing in the finite-difference time domain, Opt. Lett. 31 (20) (2006) 2972.

[6] Z. Chen, J. Xu, The generalized TLM-based FDTD modeling of frequency-dependent and anisotropic media, IEEE Trans. Microwave Theory Tech. 46 (5) (1998) 561.

[7] A.P. Zhao, J. Juntunen, A.V. Räisänen, A generalized compact 2-D FDTD model for the analysis of guided modes of anisotropic waveguides with arbitrary tensor permittivity, Microwave Opt. Technol. Lett. 18 (1) (1998) 17.

[8] J.A. Pereda, L.A. Vielva, A. Vegas, A. Prieto, A treatment of magnetized ferrites using the FDTD method, IEEE Microwave Guided Wave Lett. 3 (5) (1993) 136.

[9] F. Hunsberger, R. Luebbers, K. Kunz, Finite-difference time-domain analysis of gyrotropic media – I: magnetized plasma, IEEE Trans. Antennas Propag. 40 (12) (1992) 1489.

[10] N. Kaneda, B. Houshmand, T. Itoh, FDTD analysis of dielectric resonators with curved surfaces, IEEE Trans. Microwave Theory Tech. 45 (9) (1997) 1645.

[11] S. Dey, R. Mittra, A conformal finite-difference time-domain technique for modeling cylindrical dielectric resonators, IEEE Trans. Microwave Theory Tech. 47 (9) (1999) 1737.

[12] W. Yu, R. Mittra, A conformal finite difference time domain technique for modeling curved dielectric surfaces, IEEE Microwave Wirel. Compon. Lett. 11 (1) (2001) 25.

[13] J.-Y. Lee, N.-H. Myung, Locally tensor conformal FDTD method for modeling arbitrary dielectric surfaces, Microwave Opt. Technol. Lett. 23 (4) (1999) 245.

[14] T.I. Kosmanis, T.D. Tsiboukis, A systematic and topologically stable conformal finite-difference time-domain algorithm for modeling curved dielectric interfaces in three dimensions, IEEE Trans. Microwave Theory Tech. 51 (3) (2003) 839.

[15] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Plannery, Numerical Recipes in C++: The Art of Scientific Computing, second ed., Cambridge University Press, New York, 2002.

[16] I.A. Zagorodnov, R. Schuhmann, T. Weiland, A uniformly stable conformal FDTD-method in Cartesian grids, Int. J. Numer. Model. 16 (2) (2003) 127.

[17] C. Nieter, J.R. Cary, VORPAL: a versatile plasma simulation code, J. Comput. Phys. 196 (2) (2004) 448.

[18] J.D. Joannopoulos, Photonic Crystals: Molding the Flow of Light, Princeton University Press, Princeton, NJ, 1995.

[19] S.G. Johnson, J.D. Joannopoulos, Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis, Opt. Exp. 8 (2001) 173.

[20] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, Math. Comput. 29 (130) (1975) 396.

[21] M.S. Puckett, J.C. Brown, Accuracy of frequency estimates using the phase vocoder, IEEE Trans. Speech Audio Process. 6 (2) (1998) 166.

[22] V.A. Mandelshtam, H.S. Taylor, Harmonic inversion of time signals and its applications, J. Chem. Phys. 107 (17) (1997) 6756, Erratum: J. Chem. Phys. 109(10) (1998) 4128.