

Journal of Computational Acoustics
© IMACS

AN ACOUSTIC FINITE-DIFFERENCE TIME-DOMAIN ALGORITHM WITH ISOTROPIC DISPERSION

CHRISTOPHER L. WAGNER

*School of Electrical Engineering and Computer Science
Washington State University
Pullman, Washington 99164-2752, USA
clwagner@eecs.wsu.edu*

JOHN B. SCHNEIDER

*School of Electrical Engineering and Computer Science
Washington State University
Pullman, Washington 99164-2752, USA
schneidj@eecs.wsu.edu*

Received ()

Revised ()

The classic Yee Finite-Difference Time-Domain (FDTD) algorithm employs central differences to achieve second-order accuracy, i.e., if the spatial and temporal step sizes are reduced by a factor of n , the phase error associated with propagation through the grid will be reduced by a factor of n^2 . The Yee algorithm is also second-order isotropic meaning the error as a function of the direction of propagation has a leading term which depends on the square of the discretization step sizes. An FDTD algorithm is presented here that has second-order accuracy but fourth-order isotropy. This algorithm permits a temporal step size 50 percent larger than that of the three-dimensional Yee algorithm. Pressure-release resonators are used to demonstrate the behavior of the algorithm and to compare it with the Yee algorithm. It is demonstrated how the increased isotropy enables post-processing of the simulation spectra to correct much of the dispersion error. The algorithm can also be optimized at a specified frequency, substantially reducing numerical errors at that design frequency. Also considered are simulations of scattering from penetrable spheres ensonified by a pulsed plane wave. Each simulation yields results at multiple frequencies which are compared to the exact solution. In general excellent agreement is obtained.

Keywords: FDTD methods; transient radiation and scattering; volume scattering.

1. Introduction

Finite-difference time-domain (FDTD) algorithms assume a discretization of the given problem space and use finite-differences to approximate the derivatives in the governing differential equations. Using these approximate equations, one can express future fields in terms of known past fields. A simulation is obtained by marching the fields forward in time. However, because of the approximations inherent in the FDTD equations, fields accumulate errors as they propagate. The amount of error is dependent on the direction of propagation and the frequency, i.e., the error is both anisotropic and dispersive.

The Yee FDTD algorithm is arguably the oldest, simplest, and most common FDTD technique

(e.g., 1–6). It employs standard central-differences to approximate both the spatial and temporal derivatives of the governing equations. These central differences yield second-order accuracy in terms of propagation in a homogeneous grid.⁷ Thus if the discretization is reduced by a factor of n , the phase error will be reduced by a factor of n^2 . The Yee algorithm is also second-order in terms of isotropy, i.e., the error as a function of angle has a leading term that depends on the square of the discretization size. For “large” problems (or ones with large path lengths which include problems with relatively small geometries but multiple reflections), the error that accumulates as the field propagates over the entire domain may be unacceptable. Because of memory and other computer limitations, one often cannot simply increase the discretization to achieve an acceptable level of error. Instead, a different FDTD algorithm can be employed which has superior error characteristics. Several such algorithms have been proposed in the electromagnetics literature and a survey of many of the more promising ones can be found in ref. 8.

Here the electromagnetic algorithm proposed by Forgy^{9–11} is used to motivate the development of a new acoustic FDTD technique that is designed to improve the isotropy and stability limit relative to the Yee algorithm. In order to formalize the analysis of the algorithm, a finite-difference calculus is presented in Section 2. Using this calculus the algorithm is defined in Section 3 and the dispersion relation is obtained in Section 4. The new algorithm is shown to have second-order accuracy but fourth-order isotropy. Despite the fact that the algorithm has the same order of accuracy as the Yee algorithm, the new algorithm permits a larger temporal step and the resulting phase error is lower than that of the Yee algorithm. Additionally, because of the increased isotropy, it is possible to “tune” the coefficients of the algorithm in order to eliminate most algorithmic dispersion error at a specific design frequency, as will be discussed in Section 5. Another type of correction made possible by the increased isotropy is discussed in Section 6 where it is demonstrated that post-processing of homogeneous-domain spectra can correct much of the dispersion error over the entire spectrum. Section 7 demonstrates the behavior of the algorithm and the use of optimization by considering modes in resonators with Dirichlet boundaries. Finally, Section 8 presents simulations of plane-wave scattering from penetrable spheres. Results at multiple frequencies are compared to the exact solution.

2. Finite-Difference Calculus

In order to specify precisely a finite-difference algorithm, it is convenient to define the following discrete operators. We inherently assume a Cartesian grid of nodes at which samples of the fields are available. The shift operator $S_n(a)$ acting on a field $f(x, y, z, t)$ shifts the variable n of f by $a\delta_n$, where δ_n is the grid spacing in the n direction. Thus, for example, $S_x(a)f(x, y, z, t) = f(x + a\delta_x, y, z, t)$. With this shift operator the central finite-differences are

$$D_n = \frac{S_n(\frac{1}{2}) - S_n(-\frac{1}{2})}{\delta_n}, \quad n \in (x, y, z, t). \quad (1)$$

For the present algorithm the following averaging operators are also needed

$$A_n = \frac{S_n(1) + S_n(-1)}{2}, \quad n \in (x, y, z). \quad (2)$$

When finding the dispersion relation and the stability limit below, the effect of these operators on plane waves is required. Given a scalar plane wave, $P(x, y, z, t) = e^{j(\mathbf{k}\cdot\mathbf{r}-\omega t)}$, the difference operators acting on the wave yield

$$\begin{aligned} D_n P &= \frac{2i}{\delta_n} \sin\left(\frac{k_n \delta_n}{2}\right) P, \quad n \in (x, y, z) \\ D_t P &= -\frac{2i}{\delta_t} \sin\left(\frac{\omega \delta_t}{2}\right) P. \end{aligned} \quad (3)$$

When acting on a plane wave the averaging operators yield

$$A_n P = \cos(k_n \delta_n) P, \quad n \in (x, y, z). \quad (4)$$

In this notation the Yee Del operator (∇_0) is given by

$$\nabla_0 = D_x \hat{\mathbf{x}} + D_y \hat{\mathbf{y}} + D_z \hat{\mathbf{z}}. \quad (5)$$

The Yee gradient of a scalar plane wave is then

$$\begin{aligned} \nabla_0 P(x, y, z) &= (D_x \hat{\mathbf{x}} + D_y \hat{\mathbf{y}} + D_z \hat{\mathbf{z}}) P(x, y, z) \\ &= \left(\frac{2i}{\delta_x} \sin\left(\frac{k_x \delta_x}{2}\right) \hat{\mathbf{x}} + \frac{2i}{\delta_y} \sin\left(\frac{k_y \delta_y}{2}\right) \hat{\mathbf{y}} + \frac{2i}{\delta_z} \sin\left(\frac{k_z \delta_z}{2}\right) \hat{\mathbf{z}} \right) P(x, y, z). \end{aligned} \quad (6)$$

The standard Yee divergence operator uses the six velocity components nearest to the pressure node being updated. Using the above notation this divergence becomes

$$\begin{aligned} \nabla_0 \cdot \mathbf{V} &= D_x V_x + D_y V_y + D_z V_z \\ &= \frac{2i}{\delta_x} \sin\left(\frac{k_x \delta_x}{2}\right) V_x + \frac{2i}{\delta_y} \sin\left(\frac{k_y \delta_y}{2}\right) V_y + \frac{2i}{\delta_z} \sin\left(\frac{k_z \delta_z}{2}\right) V_z, \end{aligned} \quad (7)$$

where \mathbf{V} is a vector plane wave.

3. The Constructed Divergence

The divergence operator presented here is made by a linear combination of the six nodes which are used in the Yee algorithm (two in each grid axis direction) and the 24 next nearest neighbors. Thus this divergence operator has 30 velocity nodes contributing to the calculation. The constructed algorithm divergence operator can be represented by

$$\alpha_0 \nabla_0 + \alpha_1 \nabla_1, \quad (8)$$

where the α 's are the combination weights (α_0 is the Yee weight), ∇_0 is given in Eq. (5), and ∇_1 is

$$\nabla_1 = D_x \left(\frac{A_y + A_z}{2} \right) \hat{\mathbf{x}} + D_y \left(\frac{A_x + A_z}{2} \right) \hat{\mathbf{y}} + D_z \left(\frac{A_x + A_y}{2} \right) \hat{\mathbf{z}}. \quad (9)$$

One component of the ∇_1 operator averages four nodes, then takes the finite difference of the average of the opposite face, for example

$$\nabla_1 \cdot \hat{\mathbf{x}} = D_x \frac{A_y + A_z}{2} = \frac{(S_x(\frac{1}{2}) - S_x(-\frac{1}{2}))(S_y(1) + S_y(-1) + S_z(1) + S_z(-1))}{4\delta_x}. \quad (10)$$

4 Wagner and Schneider

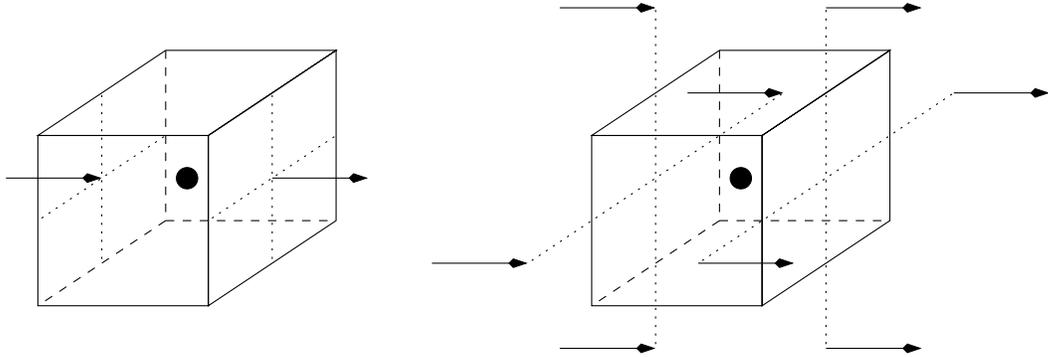


Fig. 1. The diagram on the left shows the two velocity nodes which contribute to the x component of the Yee divergence, $\nabla_0 \cdot \mathbf{V}$. The diagram on the right shows the eight velocity nodes which contribute to the x component of the $\nabla_1 \cdot \mathbf{V}$ divergence.

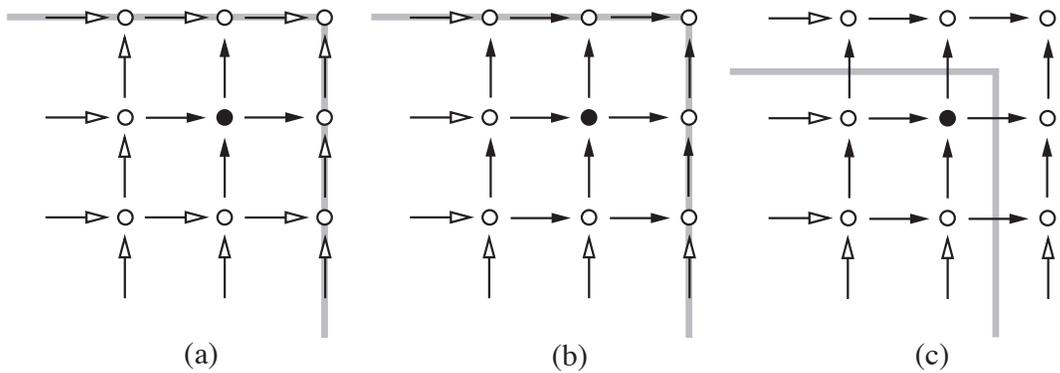


Fig. 2. Two dimensional slice of the stencil of the (a) Yee and (b) constructed divergence operators, acting in the upper right corner of a domain. The hollow circles are pressure nodes, the solid circle is the pressure node being updated by the divergence operator. The arrows are velocity nodes: filled arrowheads represent nodes used in the update of the pressure node. A rigid boundary is represented in (c) where velocity nodes on the boundary, or beyond it, would be set to zero. See the text for details.

The geometry implicit in Eq. (10) is shown in Fig. 1. If $\alpha_0 = 1$ and $\alpha_1 = 0$ in Eq. (8), then the Yee divergence is obtained. Even though the constructed divergence operator uses more velocity nodes than the Yee divergence, the stencil of the constructed operator is compact. This means that the constructed divergence operator can be applied as close to the domain walls or corners as the Yee operator.

The compact nature of the Yee and constructed operators are illustrated in Fig. 2 which shows how these divergence operators, which use \mathbf{V} nodes to compute a P node, can both be used equally close to the domain corners. In Fig. 2 the pressure node being updated is indicated by a filled circle. The neighboring velocity nodes used in the update equation are shown with filled arrowheads. The Yee algorithm is depicted in Fig. 2(a) while the proposed algorithm is shown in Fig. 2(b). Later, resonators satisfying a Dirichlet (pressure-release) boundary condition will be considered. In that

case one merely sets to zero the pressure nodes on the boundary (such as might correspond to the gray boundary indicated in the figure). The velocity nodes tangential to the boundary are zero since they are only updated using pressures which are explicitly zero owing to the boundary condition. A Neumann (rigid) boundary can be realized by setting the normal component of velocity to zero at the boundary. This is depicted in Fig. 2(c) where the gray boundary has been shifted to correspond to the effective location of a rigid boundary. In this case the velocity nodes which are intersected by, or beyond, the boundary are explicitly set to zero but no other special considerations need be used to apply the proposed algorithm. One consequence of the spatial stencil for the proposed algorithm is that rigid objects must be at least one cell thick (whereas in the Yee algorithm a rigid sheet could be realized by setting to zero velocity nodes over a single plane). For inhomogeneous media, the proposed algorithm behaves the same as the Yee algorithm where coefficients are dictated by the density and sound speed which pertain at the node being updated.

4. Dispersion Relation and Stability Limit

The small-signal acoustic governing equations for velocity and pressure are

$$\frac{\partial \mathbf{V}}{\partial t} = -\frac{1}{\rho} \nabla P, \quad (11)$$

$$\frac{\partial P}{\partial t} = -\rho c^2 \nabla \cdot \mathbf{V} + P_{\text{source}}, \quad (12)$$

where \mathbf{V} is the velocity, P is the pressure, c and ρ are the wave speed and density respectively, and P_{source} is an additive pressure source. The space and time derivative symbols are to be interpreted according to the context. They may refer to continuum operators or to discrete operations on the grid. For the second-order constructed algorithm only Eq. (12) is different from the Yee FDTD update. To include the constructed operator's extra nodes Eq. (12) is written with Eq. (8) for the divergence operator

$$D_t P = -\rho c^2 (\alpha_0 \nabla_0 \cdot \mathbf{V} + \alpha_1 \nabla_1 \cdot \mathbf{V}) + P_{\text{source}}. \quad (13)$$

Substituting the Yee implementation of Eq. (11) into the time derivative of Eq. (13) and assuming a source-free region produces the wave equation for the algorithm

$$D_t^2 P = c^2 (\alpha_0 \nabla_0 \cdot \nabla_0 + \alpha_1 \nabla_1 \cdot \nabla_0) P. \quad (14)$$

Substituting the values of the operators from Eqs. (3) and (4) into Eq. (14) and simplifying, produces the dispersion relation for the algorithm

$$\begin{aligned} \left(\frac{2}{c\delta_t}\right)^2 \sin^2\left(\frac{\omega\delta_t}{2}\right) &= \left(\frac{2}{\delta}\right)^2 \left[\sin^2\left(\frac{k_x\delta}{2}\right) \left(\alpha_0 + \frac{\alpha_1}{2} (\cos(k_y\delta) + \cos(k_z\delta))\right) \right. \\ &\quad + \sin^2\left(\frac{k_y\delta}{2}\right) \left(\alpha_0 + \frac{\alpha_1}{2} (\cos(k_x\delta) + \cos(k_z\delta))\right) \\ &\quad \left. + \sin^2\left(\frac{k_z\delta}{2}\right) \left(\alpha_0 + \frac{\alpha_1}{2} (\cos(k_x\delta) + \cos(k_y\delta))\right) \right], \end{aligned} \quad (15)$$

where the δ is the cubic-cell grid size, and k_x, k_y, k_z , are the wave number components in the x, y , and z directions respectively. In Eq. (15) the k 's and ω may be exact (i.e., continuum or theoretical) or numerical (grid), depending on context. For example, in the resonators considered in Section 7, the modes dictate that the wave numbers for a supported mode are exact, which are then used in Eq. (15) to find the actual numerical frequencies. Henceforth a tilde will be used to indicate numerical frequencies and wave numbers, so $\tilde{\omega}$ and \tilde{k} indicate numerical values, ω and k indicate exact (theoretical) values.

The algorithm weights have not been specified. The α 's are chosen to obtain fourth-order isotropy. Expanding the right hand side of the dispersion relation Eq. (15) in a power series in the space step size δ , gives

$$(\alpha_0 + \alpha_1)(k_x^2 + k_y^2 + k_z^2) - \frac{1}{12}(\alpha_0(k_x^4 + k_y^4 + k_z^4) + \alpha_1(k_x^4 + k_y^4 + k_z^4 + 6k_x^2k_y^2 + 6k_x^2k_z^2 + 6k_y^2k_z^2))\delta^2 + O[\delta]^4. \quad (16)$$

We now require $\alpha_0 + \alpha_1 = 1$ to normalize the leading term to the desired exact result

$$(k_x^2 + k_y^2 + k_z^2) - \frac{1}{12}(k_x^4 + 6\alpha_1k_y^2k_z^2 + k_y^4 + 6\alpha_1k_x^2k_z^2 + k_z^4 + 6\alpha_1k_x^2k_y^2)\delta^2 + O[\delta]^4. \quad (17)$$

Using $k^2 = k_x^2 + k_y^2 + k_z^2$ and letting α_1 be $\frac{1}{3}$ yields

$$k^2 - \frac{k^4\delta^2}{12} + O[\delta]^4. \quad (18)$$

Thus the operator is second-order accurate for the derivative, and fourth-order isotropic. The isotropy is fourth-order because the second-order error term is dependent on k , without explicit dependence on k_x, k_y , or k_z . The additional nodes used by the operator do not increase the order of accuracy of the derivative. With α_1 set to $1/3$, $\alpha_0 = 2/3$ immediately follows.

To find the Courant stability limit the dispersion relation Eq. (15) is solved for the frequency

$$\omega = \frac{2}{\delta_t} \sin^{-1} \left(S \left[\sin^2 \left(\frac{k_x \delta}{2} \right) \left(\alpha_0 + \frac{\alpha_1}{2} (\cos(k_y \delta) + \cos(k_z \delta)) \right) \right. \right. \\ \left. \left. + \sin^2 \left(\frac{k_y \delta}{2} \right) \left(\alpha_0 + \frac{\alpha_1}{2} (\cos(k_x \delta) + \cos(k_z \delta)) \right) \right. \right. \\ \left. \left. + \sin^2 \left(\frac{k_z \delta}{2} \right) \left(\alpha_0 + \frac{\alpha_1}{2} (\cos(k_x \delta) + \cos(k_y \delta)) \right) \right]^{1/2} \right), \quad (19)$$

where $S = c\delta_t/\delta$ is the Courant number. The Courant stability limit can be found using complex frequency analysis^{12,13} where stability requires that ω be real for all real k 's permitted by grid sampling. Thus, in Eq. (19), the argument of the square-root must be positive, and the magnitude of the argument of the arcsine must be less than or equal to one. Typically the limiting case will occur when one or more of k_x, k_y , and k_z are equal to π/δ . For $\alpha_1 \leq 1/4$ the limiting case is when $k_x = k_y = k_z = \pi/\delta$, and when $1/4 < \alpha_1 \leq 1/2$ the limiting case is when two k 's are π/δ and

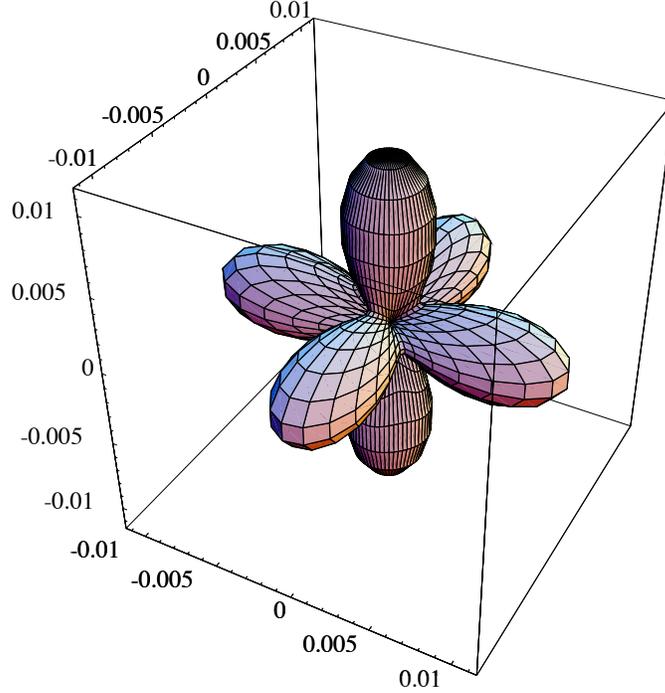


Fig. 3. Dispersion error surface for the Yee algorithm at 10 samples per wavelength. The Courant number is $1/\sqrt{3}$ and $\alpha_1 = 0$.

one is zero. Using these conditions the Courant limit is as follows

$$S = \begin{cases} \frac{1}{\sqrt{3}} & \text{for } \alpha_1 = 0, \text{ the Yee algorithm,} \\ \frac{1}{\sqrt{3(1-2\alpha_1)}} & \text{for } 0 \leq \alpha_1 \leq \frac{1}{4}, \\ \frac{1}{\sqrt{2(1-\alpha_1)}} & \text{for } \frac{1}{4} \leq \alpha_1 \leq \frac{1}{2}, \\ \frac{\sqrt{3}}{2} & \text{for } \alpha_1 = \frac{1}{3}, \text{ the present isotropic algorithm,} \\ 1 & \text{for } \alpha_1 = \frac{1}{2}, \text{ maximum } \alpha_1 \text{ for stable algorithm.} \end{cases} \quad (20)$$

To illustrate the effect of the constructed operator on the FDTD algorithm Figs. 3 and 4 show the phase velocity error surfaces for the Yee and the present algorithm. Both figures are plots of $1 - \tilde{\omega}/\omega$ at $k = 2\pi/(10\delta)$ (i.e., 10 cells per wavelength), when using the respective Courant limits. The quantity $1 - \tilde{\omega}/\omega$ is a measure of the phase error at a given wave number. The ideal surface has zero radius, the dispersion error in a given direction is proportional to the radial distance from the origin. The enhanced isotropy of the constructed algorithm is evident because the error surface is nearly spherical.

Although the proposed algorithm effectively uses different spatial nabla operators for the update of the pressure and velocity, it is worth noting the same dispersion relation is obtained whether ones solves for the velocity or the pressure. Hence both fields have the same level of accuracy.

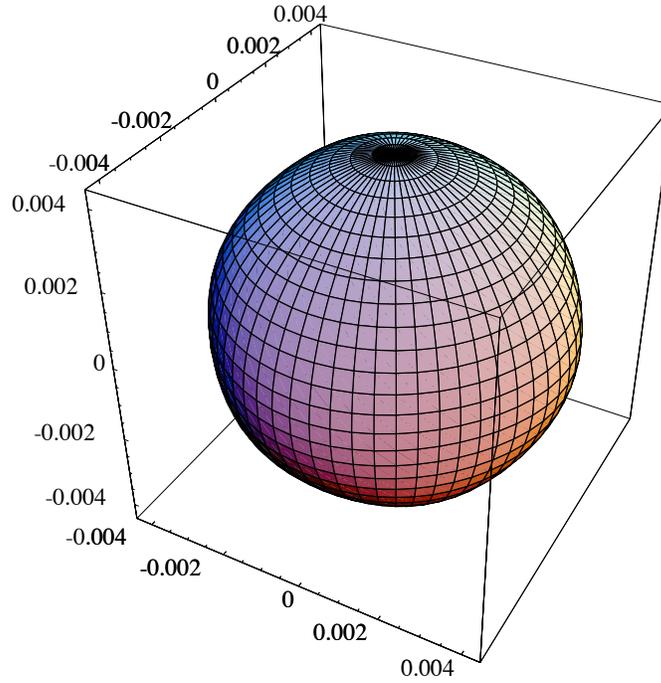


Fig. 4. Dispersion error surface for the present algorithm measured at 10 samples per wavelength. Note the smaller scale factors compared to Fig. 3. The Courant number is $\sqrt{3}/2$ and $\alpha_1 = 1/3$.

5. Run-Time Frequency Optimization

It is possible to “tune” this algorithm by adjusting the α_0 and α_1 coefficients to improve the run-time performance at a particular frequency. The method is based on the technique described by Forgy.¹⁰ Optimization at a particular frequency involves setting the α_0 and α_1 coefficients to eliminate most of the error in the spatial derivative at the design frequency. Time optimization is performed by interpreting the simulation data time series as having a reduced time step δ_t , even though the simulation is still run at the δ_t found from the stability limit in Eq. (20).

To optimize the spatial derivative the dispersion error in two different directions are made equal by adjusting the α 's. When $\alpha_1 = 0$ (i.e., the Yee algorithm) the worst case dispersion error is along the grid axis. When $\alpha_1 = 1/2$ the worst case dispersion error is along the major diagonals. Given a design wave number k_{opt} with corresponding design frequency $\omega_{\text{opt}} = k_{\text{opt}}/c$, we equate the spatial dispersion along the grid axis

$$\left(\frac{2}{\delta}\right)^2 (\alpha_0 + \alpha_1) \sin^2\left(\frac{k_{\text{opt}}\delta}{2}\right), \quad (21)$$

with the spatial dispersion along the major diagonal

$$\frac{12}{\delta^2} \left[\alpha_0 + \alpha_1 \cos\left(\frac{k_{\text{opt}}\delta}{\sqrt{3}}\right) \right] \sin^2\left(\frac{k_{\text{opt}}\delta}{2\sqrt{3}}\right), \quad (22)$$

which produces

$$(\alpha_0 + \alpha_1) \sin^2\left(\frac{k_{\text{opt}}\delta}{2}\right) = 3 \left[\alpha_0 + \alpha_1 \cos\left(\frac{k_{\text{opt}}\delta}{\sqrt{3}}\right) \right] \sin^2\left(\frac{k_{\text{opt}}\delta}{2\sqrt{3}}\right). \quad (23)$$

The spatial optimization is performed by simultaneously solving Eq. (23) and $\alpha_0 + \alpha_1 = 1$ for α_0 and α_1 giving

$$\alpha_0 = (1 - \alpha'), \quad \alpha_1 = \alpha', \quad (24)$$

where

$$\alpha' = \frac{2 + \cos(k_{\text{opt}}\delta) - 3 \cos\left(\frac{k_{\text{opt}}\delta}{\sqrt{3}}\right)}{12 \sin^4\left(\frac{k_{\text{opt}}\delta}{2\sqrt{3}}\right)}. \quad (25)$$

The optimization to this point ensures that the dispersion-shifted frequencies $\tilde{\omega}$ found via Eq. (19) are equal for propagation along the grid axes and major diagonal directions at the design wave number k_{opt} . These α 's, i.e., Eq. (24), give a specific algorithm, the stability limit of which is given by Eq. (20). Once the stability limit is found a time step δ_t for the algorithm can be chosen.

Now the time optimization is performed by finding an effective temporal step δ_t' using the dispersion relation Eq. (19) (where the frequency is the numeric frequency $\tilde{\omega}$) with parameters given in Eq. (24) along the grid axis direction, i.e.,

$$\delta_t' = \frac{\tilde{\omega}}{ck_{\text{opt}}}\delta_t = \frac{2}{ck_{\text{opt}}}\sin^{-1}\left(S \sin\left(\frac{k_{\text{opt}}\delta}{2}\right)\right). \quad (26)$$

This effective temporal step is used in interpreting the time-series data from the simulation. This reinterpretation shifts the frequency of the optimized k to the correct ω . The simulation is still run with the δ_t found above from stability considerations. The dispersion error surface for the optimized algorithm is shown in Fig. 5. Although this error surface now appears anisotropic, one must note the small level of error. This surface essentially represents the deviations from a perfect sphere for the surface shown in Fig. 4.

6. Post-Processing Frequency Correction

In addition to the run-time tuning of the algorithm above, isotropic algorithms in a homogeneous space permit the post-processing of the spectra from a simulation to correct the dispersion errors (shifted frequencies). The algorithm here is isotropic to fourth order, so frequency correction can be performed where the goal is essentially to eliminate the mean dispersion error at each frequency. The method uses the dispersion relation Eq. (15) to perform the correction, where one solves for k given an observed frequency ω' . The corrected frequency is then given by $\omega = ck(\omega')$. (It should be noted that one can attempt to correct for the mean dispersion error in the Yee algorithm as was done by Nehrbass *et al.*¹⁴ However, because of the second-order nature of the isotropy, this can only provide modest improvements as is discussed further in the next section.)

In solving Eq. (15) for k , a direction of propagation must be assumed. In a truly isotropic algorithm, any choice of direction would be equivalent. In the present algorithm, the (1, 1, 1) major

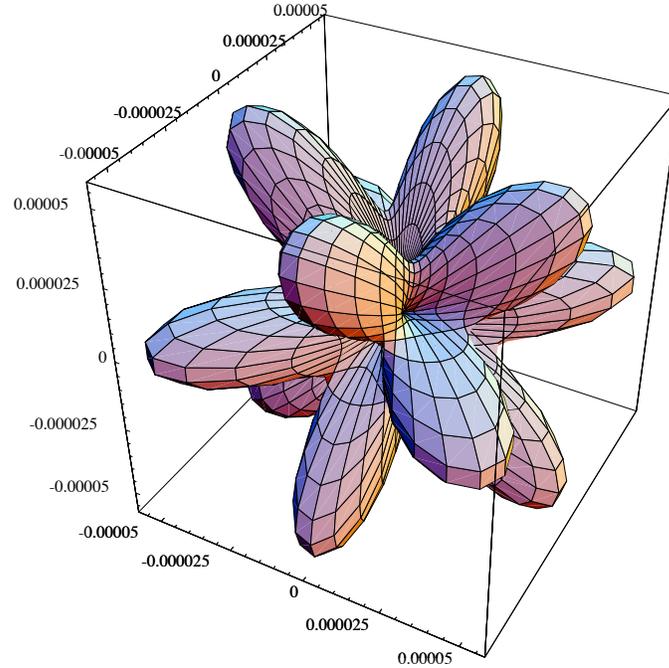


Fig. 5. Dispersion error surface for the optimized algorithm at 10 samples per wavelength. $\alpha'_0 = 0.6651999117$, $\alpha'_1 = 0.3348000883$, and $S = 0.866979664$.

diagonal direction is close to the direction of the average dispersion error, so it is used as the correction direction in the results presented in the following section. To restate the correction procedure, an observed frequency ω' in a simulation spectra is used in the left of the dispersion relation Eq. (15). Then, given a direction, the right-hand side is solved for the magnitude of the wavenumber, i.e., $k(\omega')$. For an arbitrary direction this would require that the equation be solved numerically (e.g., by the bisection algorithm). However when propagation is assumed along the grid diagonal of a uniform grid, such that $k_x = k_y = k_z = k/\sqrt{3}$ and the spatial step is δ in all directions, an analytic expression can be obtained for k :

$$k(\omega') = \frac{2\sqrt{3}}{\delta} \sin^{-1} \left(\frac{1}{2} \sqrt{3 - \sqrt{9 - \frac{8}{S^2} \sin^2 \left(\frac{\omega' \delta_t}{2} \right)}} \right). \quad (27)$$

Multiplying Eq. (27) by the sound speed c gives the corrected frequency for which the phenomena observed at ω' should be assumed to have occurred. Such a correction is demonstrated in the following section.

Depending on the direction and the frequency, the wavenumber obtained from Eq. (27) may be complex.¹⁵ The correction process is terminated when k is complex. This will happen when the argument of the innermost radical is negative and only occurs at the uppermost frequencies of the simulation. For example, using the Courant limit of $\sqrt{3}/2$, the inner-most radical is zero when

$\sin(\omega\delta_t/2) = \sqrt{27/32}$. Since $\omega\delta_t/2$ is equivalent to $\pi S/N_\lambda$, where N_λ is the number of cells per wavelength, the radical is zero when the discretization is such that there are approximately 2.337 cells per wavelength of propagation, i.e., an extremely high frequency in terms of standard FDTD discretizations. For frequencies beyond this value (or discretization that are coarser than this value), $k(\omega')$ will be complex and hence one cannot obtain a corrected frequency for ω .

7. FDTD Simulations: Resonators

To compare and contrast this algorithm with the Yee algorithm, pressure-release boundary (Dirichlet, $P = 0$) cubic resonators are used. Thus no absorbing boundary is required to terminate the computational domain. The resonators are excited with an additive source in the center of the resonator and data is recorded at the same location. The resulting time series is Fourier transformed.

The resonant frequencies for a rectangular resonator are simple to find in the continuum. The symmetry and geometry of the resonator, in this case cubic with a centered source, dictates the modes that are excited. Once the k 's of the continuum modes are found, the dispersion relation Eq. (19) can be used to predict precisely the numerical frequencies of the algorithm. The same procedure can be used with the Yee dispersion relation.

A cubic resonator with pressure-release boundary walls has resonant mode frequencies given by

$$\omega^2 = c^2 \left[\left(\frac{m\pi}{L} \right)^2 + \left(\frac{n\pi}{L} \right)^2 + \left(\frac{p\pi}{L} \right)^2 \right], \quad (28)$$

where m , n , and p are the mode indices, and L is the size of the resonator along each axis. For the present geometry the mode indices must be odd. In terms of the wavenumber components (Eq. 28) can be written

$$\omega^2 = c^2 (k_x^2 + k_y^2 + k_z^2). \quad (29)$$

That is, given the mode indices, the corresponding wavenumbers are given by

$$k_x = \frac{m\pi}{L}, \quad k_y = \frac{n\pi}{L}, \quad k_z = \frac{p\pi}{L}. \quad (30)$$

The numerical frequencies that will exist in the FDTD simulation can be found¹⁶ from Eq. (19), with the k 's from Eq. (30)

$$\begin{aligned} \tilde{\omega} = \frac{2}{\delta_t} \sin^{-1} & \left(S \left[\sin^2 \left(\frac{m\pi\delta}{2L} \right) \left(\alpha_0 + \frac{\alpha_1}{2} \left(\cos \left(\frac{n\pi\delta}{L} \right) + \cos \left(\frac{p\pi\delta}{L} \right) \right) \right) \right. \right. \\ & + \sin^2 \left(\frac{n\pi\delta}{2L} \right) \left(\alpha_0 + \frac{\alpha_1}{2} \left(\cos \left(\frac{m\pi\delta}{L} \right) + \cos \left(\frac{p\pi\delta}{L} \right) \right) \right) \\ & \left. \left. + \sin^2 \left(\frac{p\pi\delta}{2L} \right) \left(\alpha_0 + \frac{\alpha_1}{2} \left(\cos \left(\frac{m\pi\delta}{L} \right) + \cos \left(\frac{n\pi\delta}{L} \right) \right) \right) \right]^{1/2} \right). \end{aligned} \quad (31)$$

Equation (31) gives the resonant frequency $\tilde{\omega}$ for a particular set of mode indices for an FDTD cubic resonator. Due to symmetry, modes with permutations of mode indices will be degenerate.

In the continuum, any resonator has an infinite number of modes. In a discrete space there will be a finite number due to the spatial sampling of the grid. The highest frequency that may be coupled into the grid, i.e., the grid Nyquist frequency, is $1/2\Delta t$.¹⁷ In the continuum there are modes whose frequencies are below the grid Nyquist frequency, but that have wavenumber components that are beyond the grid's spatial sampling limit.

The cubic resonators are excited by a single additive pressure source centered in the domain. The source is a unit amplitude pulse of duration $2\Delta t$, giving a spectral null at the grid Nyquist frequency. The pressure field is sampled at the location of the source. This data is Fourier transformed (after being windowed with a raised-cosine transform which helps emphasize the peaks) to produce the mode spectral plots shown below. With this geometry the excited and detectable modes will have odd x , y , and z mode indices. To maintain the source at the center of the domain, there must be an odd number of cells in each direction. Without loss of generality, unit cells are assigned a size of $\delta = 1$ m. The simulations use the appropriate Courant limit and are run for 65536 time steps.

For the sake of illustration, we model a fairly small resonator which is constructed from \mathfrak{P} cells. Because the walls of the resonator are constructed from pressure nodes which are set to zero, the interior length L is 8δ (the central source node has three additional non-zero pressure nodes to either side along an axis before encountering a wall). Larger resonators support more modes, but the fundamental behavior of the Yee and the proposed technique can be understood from the smaller resonator and this understanding translates to the larger structures. The speed of sound is assumed to be 1500 m/s. We consider frequencies from dc up to 600 Hz where the discretization is 2.5 cells per wavelength ($N_\lambda = 2.5$).

Figure 6 shows the magnitude of the FFT of the pressure versus frequency for the Yee algorithm. Also shown is the theoretical result which corresponds to an ideal discrete resonator that suffers no dispersion error. Each peak corresponds to one or more modes. For the theoretical (ideal) results the peaks are labeled with a triplet written horizontally which indicates the corresponding mode or modes. A label also implies the modes given by the permutations of the indices. So, for example, a label of $(1, 1, 3)$ also implies the $(1, 3, 1)$ and $(3, 1, 1)$ modes exist at the same frequency. The modes corresponding to the Yee peaks are written vertically. Note that some modes which should be degenerate, such as the $(3, 3, 3)$ and $(1, 1, 5)$ modes, are distinct in the Yee algorithm. This is a consequence of the anisotropy of the grid. The $(3, 3, 3)$ mode consists of a superposition of plane wave traveling along the grid diagonals. Such plane wave do not suffer dispersion errors in the Yee algorithm when it is run at the 3D Courant limit and hence the $(3, 3, 3)$ mode corresponds exactly to the theoretical result. However the $(1, 1, 5)$ mode (and its permutations) do not consist of plane waves propagating along the grid diagonals and hence do suffer grid dispersion with the result that this mode is closer to the $(1, 3, 3)$ mode than the $(3, 3, 3)$ mode. In addition to such mode splitting, there can be spurious degeneracies, or mode combining, such as occurs with the $(1, 1, 7)$ and $(1, 3, 5)$ modes. Further details concerning mode splitting and mode combining (as well as mode shuffling) can be found in ref. 16.

Figure 7 shows the magnitude of the FFT of the pressure versus frequency for the proposed isotropic FDTD algorithm together with the theoretic result. As before, the modes corresponding to the peaks for the theoretical data are written horizontally while they are written vertically for the FDTD results. Because of the improved isotropy of the algorithm, the effects of mode splitting

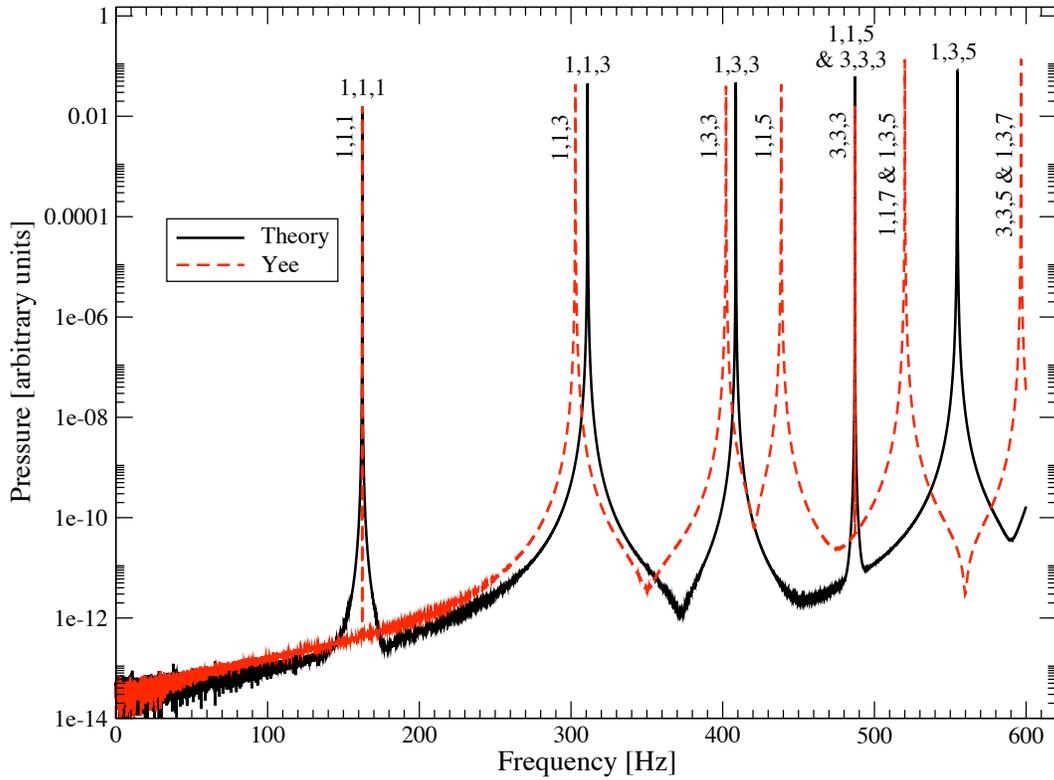


Fig. 6. Cubic resonator spectrum for Yee algorithm ($\alpha_1 = 0$) where the Courant number is $1/\sqrt{3}$ and the resonator size is $L = 8\delta$. The triplets indicate the mode or modes which correspond to a peak. The horizontal triplets are for the theory (ideal) peaks and the vertical ones for the Yee algorithm. The triplets imply all the permutations of the indices for the particular mode. Hence (1, 1, 3) also implies modes (1, 3, 1) and (3, 1, 1). The highest frequency shown, 600 Hz, corresponds to 2.5 cells per wavelength.

and mode combining are much less pronounced than in the Yee algorithm. Note that the (1, 1, 5) and (3, 3, 3) modes are still split but the separation in spectral peaks is less than two Hertz (the separation is approximately 49 Hz for the Yee algorithm). Except in the case of the diagonal modes, the proposed technique typically produces peaks which are closer to the correct value.

Figure 8 is the magnitude of the FFT of the pressure versus frequency for the proposed technique after the frequency has been corrected as described in the previous section. This correction essentially serves as a rescaling of the horizontal axis (although not a linear one). This rescaling cannot recombine modes which are distinct. Thus the (1, 1, 5) and (3, 3, 3) modes remain split, but they are mapped slightly closer to each other. Note that in the Yee algorithm, one could remove the mean dispersion error, but the large separation between the (1, 1, 5) and (3, 3, 3) modes would remain. As can be seen from Fig. 8, the corrected results agree very well with the ideal results. The only obvious discrepancy occurs for the (1, 3, 5) mode where the discretization is approximately 2.7 cells per wavelength.

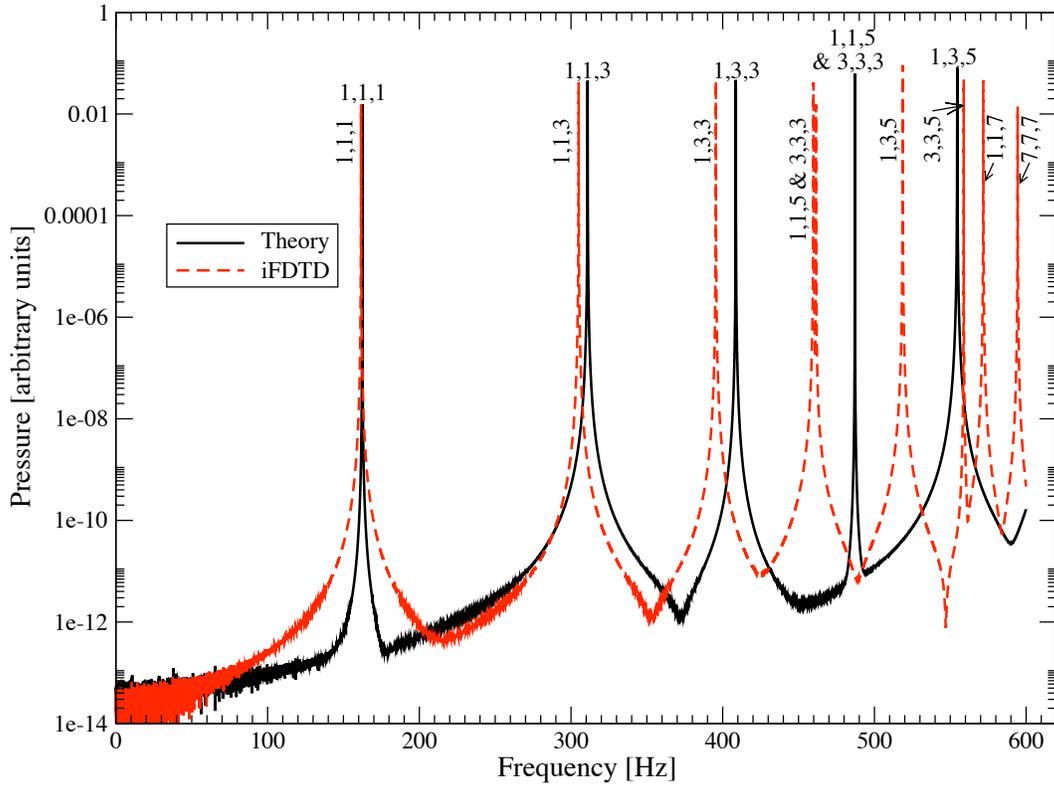


Fig. 7. Cubic resonator spectrum for proposed isotropic FDTD (iFDTD) algorithm ($\alpha_1 = 1/3$) where the Courant number is $\sqrt{3}/2$ and the resonator size is $L = 8\delta$. The horizontal triplets are for the theory (ideal) peaks and the vertical ones for the theory (ideal) peaks and the vertical ones for the iFDTD algorithm. The (1, 1, 5) and (3, 3, 3) modes are distinct in the iFDTD results but the separation is less than two Hertz and hence are labeled as a pair (the (3, 3, 3) mode is the higher of the two).

8. FDTD Simulations: Scattering

To demonstrate that the proposed algorithm works for inhomogeneous problems scattering from penetrable spheres is considered in this section. No optimization or correction is employed. The test geometry is similar to that used in ref. 18. The computational domain is $3\mathfrak{B}$ cells and the sphere is modeled using eight cells along its radius. Fig. 9 shows a two-dimensional cross-section of the computational domain where the units are in cell numbers. A plane wave is incident in the $+z$ direction. The wave is a Ricker wavelet with 20 cells per wavelength at its most energetic frequency. The incident field is introduced over a total-field/scattered-field boundary.^{19–21} The grid is terminated with a perfectly-matched layer^{3,4,22} (PML) which is not shown in Fig. 9. The PML is eight cells thick and was formulated following the coordinate-stretching approach described by Chew and Weedon.²³ Spectral information is extracted using a discrete Fourier transform.¹³ Results are compared to the exact solution over the near-field “evaluation line” which parallels the z axis and is shown in Fig. 9. Assuming unit-amplitude harmonic ensonification, the exact solution for the

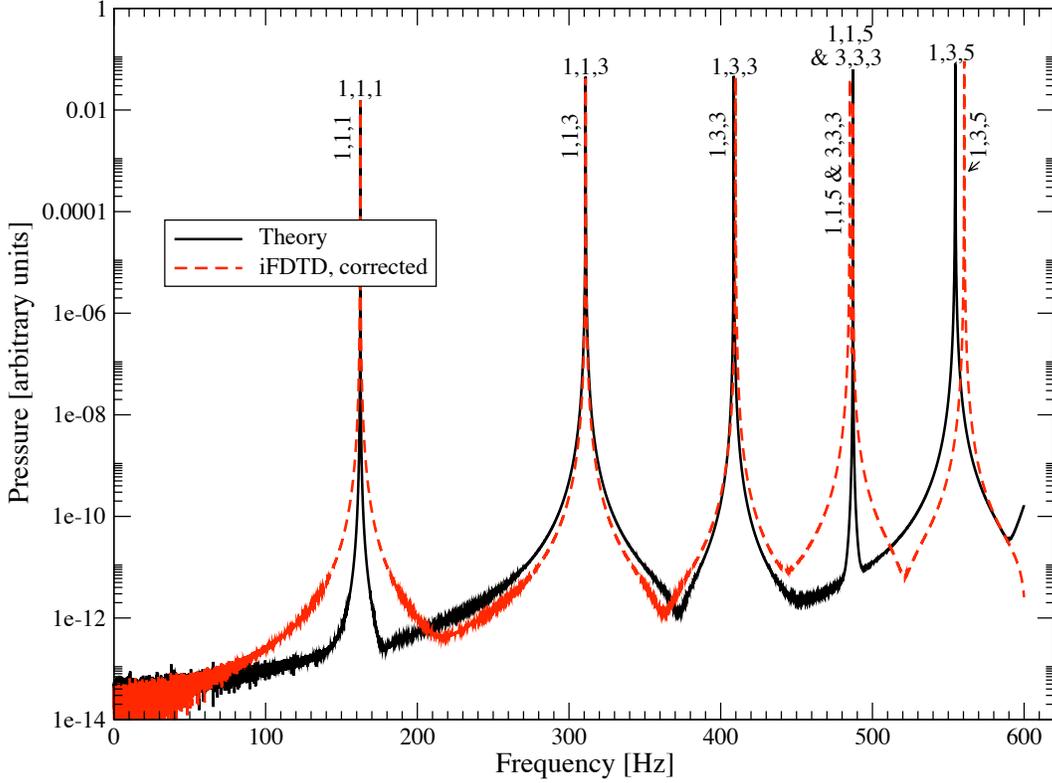


Fig. 8. Cubic resonator spectrum from Fig. 7 after applying the post-processing correction described in Sec. 6. The horizontal triplets are for the theory (ideal) peaks and the vertical ones for the iFDTD algorithm. The (1,1,5) and (3,3,3) modes are still distinct in the iFDTD results since frequency correction cannot recombine modes (but it does serve to narrow the separation between modes which are spuriously split).

scattered pressure is given by²⁴

$$P^s = - \sum_{m=0}^{\infty} (-i)^m P_m(\cos \theta) \frac{2m+1}{1+iC_m} [j_m(kR) + in_m(kR)] e^{-i\omega t}, \quad (32)$$

where j_m and n_m are the spherical Bessel and Neumann functions, respectively, P_m is the Legendre polynomial, k is the wavenumber in the background medium, R is the distance from the origin to the observation point, θ is the angle between the $-z$ direction and a ray to the observation point, and

$$C_m = \frac{j'_m(k_1 a) n(k a) - g h j_m(k_1 a) n'_m(k a)}{j'_m(k_1 a) j_m(k a) - g h j_m(k_1 a) j'_m(k a)}, \quad (33)$$

where a is the radius of the scatterer, $g = \rho_1/\rho_0$, $h = c_1/c_0$, and $k_1 = k/h$ —a subscript 0 indicates the background medium while 1 indicates the scatterer. A prime is used to indicate differentiation with respect to the argument so that $j'_m(kR) = \partial j_m(kR)/\partial(kR)$. The exact solution was calculated using *Mathematica*.²⁵

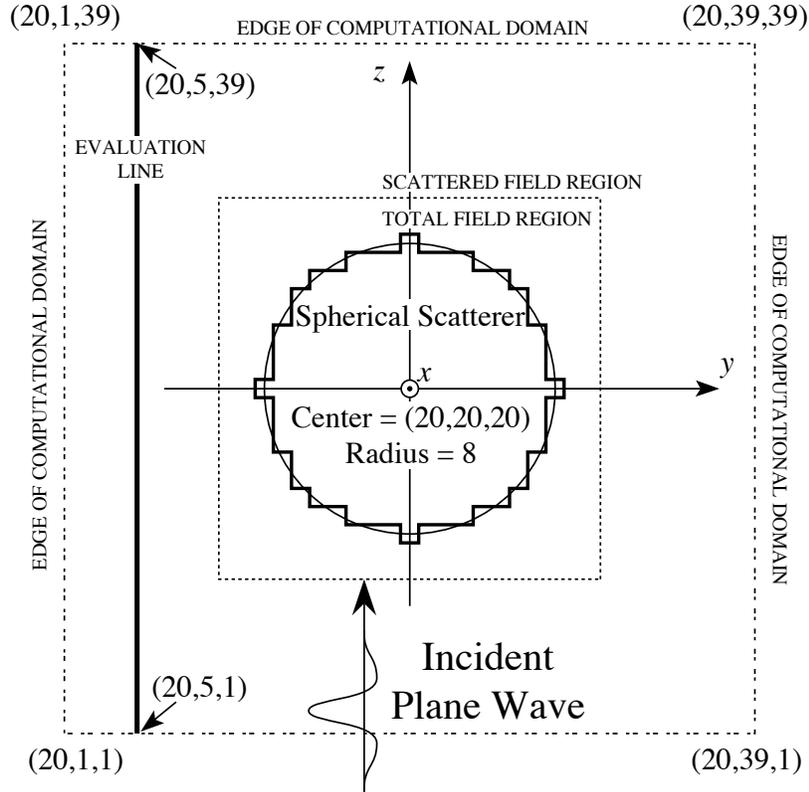


Fig. 9. Two-dimensional cross-section of the spherical scatterer and the surrounding space. The eight-cell PML used to absorb out-going waves is not shown. All units are in number of cells (thus this cross section was taken over the $x = 20$ plane).

Two separate spheres are ensonified by the pulsed source. In the first case the sound speed and the density of the scatterer are both twice that of the background medium, i.e., $q = 2c_0$ and $\rho_1 = 2\rho_0$. The simulation was performed for 512 time steps and three different frequencies corresponding roughly to those which would yield 10, 20, or 40 cells per wavelength were extracted from the measured temporal data. We have observed that when operated at their respective Courant limits both the Yee algorithm and the proposed technique may be unstable when modeling inhomogeneous media. A slight reduction of the Courant number produced stable results. Thus the Courant number employed for the following simulations was 99 percent of the limit.

Figure 10 shows the magnitude of the pressure measured over the evaluation line for the three frequencies as well as the exact solutions. Because of the discrete nature of the simulation and the Courant number which was used, the measured frequencies do not correspond to integer values of cells per wavelength. The true number of cells per wavelength are given in the legend of the figure. Note that the agreement between the measured and exact results is generally good except at the shortest wavelength where the staircasing of the surface of the scatterer has a more pronounced impact on the scattered field.

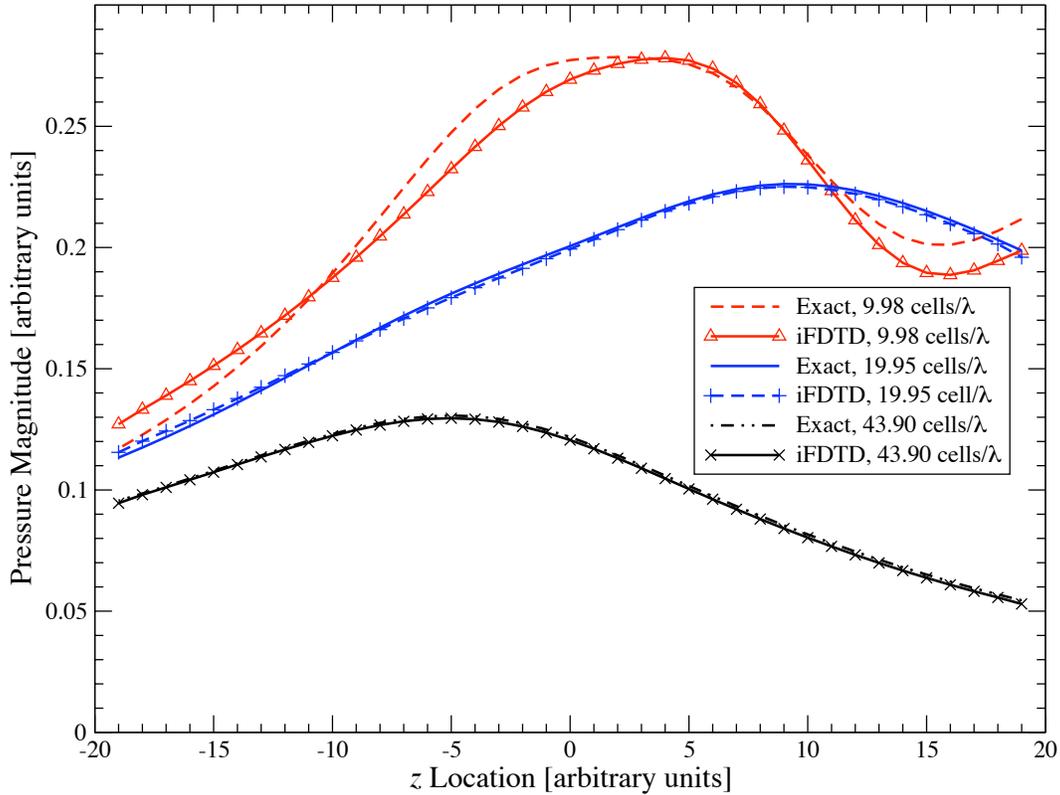


Fig. 10. Magnitude of pressure versus position at three different frequencies. The sound speed and density of the scatterer are both twice that of the background media, i.e., $c_1 = 2c_0$ and $\rho_1 = 2\rho_0$. The cells per wavelength reported in the legend are as seen in the background media. iFDTD indicates the isotropic FDTD algorithm proposed in this paper.

Figure 11 shows the scattered field when the sphere has a sound speed half that of the background medium while the density is constant everywhere, i.e., $c_1 = c_0/2$ and $\rho_1 = \rho_0$. In this case it takes much longer for the energy to ring out of the sphere so the simulation was run for 16384 time steps (again, using a Courant number 99 percent of the limit). Frequencies were recorded which correspond roughly to 10 and 20 cells per wavelength in the scatterer (and thus twice that amount in the background medium). Again the agreement between the exact and calculated results are good with some discrepancies evident at the shorter wavelength.

In both simulations no optimization was done (i.e., the coefficients were held at $\alpha_0 = 2/3$ and $\alpha_1 = 1/3$). Velocity nodes that had one neighboring pressure node in the scatterer and the other in the background medium used a density which was the average of the densities at the two adjacent pressure nodes. (This is only pertinent to the first sphere since there is no change of density for the second.)

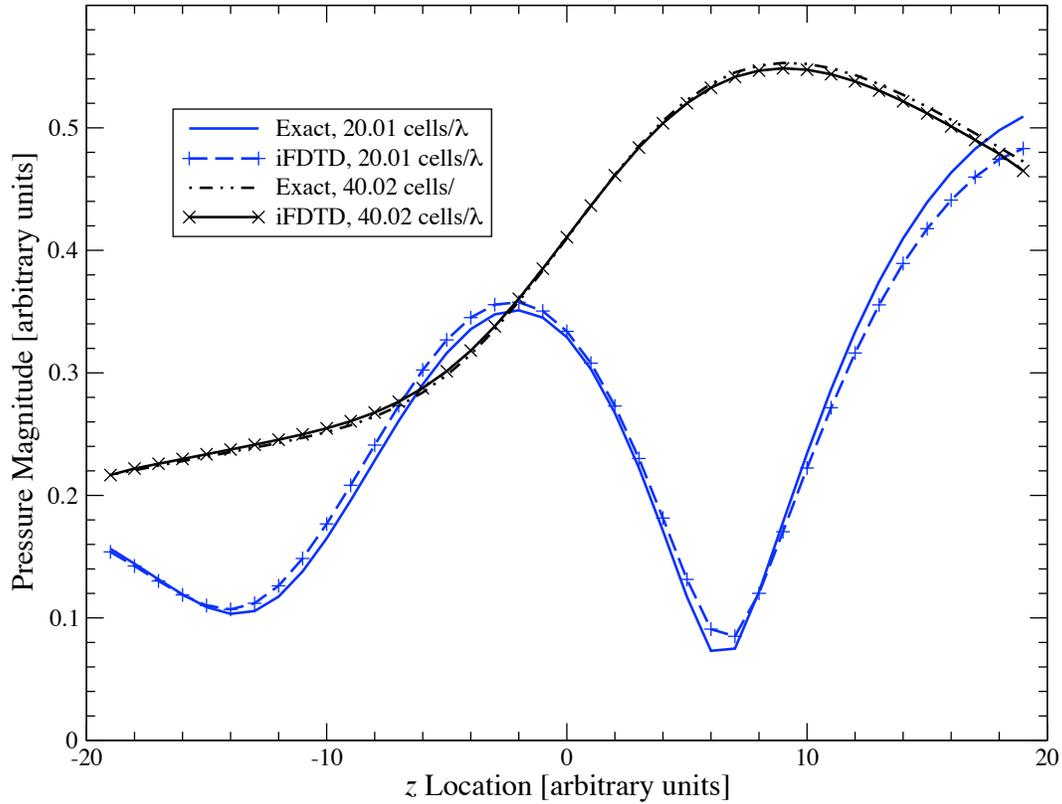


Fig. 11. Magnitude of pressure versus position at two different frequencies. The sound speed of the scatterer is half that of the background media while the density is constant everywhere, i.e., $c_1 = c_0/2$ and $\rho_1 = \rho_0$. The cells per wavelength reported in the legend are as seen in the background media (the discretization is halved within the scatterer).

9. Conclusions

The FDTD algorithm presented here is stable at $3/2$ times the usual Yee Courant limit. In addition the operator is isotropic to fourth order. Isotropy improves the mode structure behavior over the frequencies which are typically of interest. Isotropic (or nearly isotropic) algorithms (in homogeneous regions) can be post-processed to reduce much of the dispersion error induced frequency shifting. The algorithm was shown to yield good results for canonical scattering problems.

Acknowledgments

Funding for this work was provided by the Harold and Diana Frank EECS Graduate Fellowship and the Office of Naval Research, Code 3210A.

References

1. K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. Antennas Propagat.*, 14(3):302–307, March 1966.
2. J. LoVetri, D. Mardare, and G. Soulodre. Modeling of the seat dip effect using the finite-difference time-domain method. *J. Acoust. Soc. Am.*, 100(4):2204–2212, October 1996.
3. J. G. Maloney and K. E. Cummings. Adaptation of FDTD techniques to acoustic modeling. In *11th Annual Review of Progress in Applied Computational Electromagnetics*, volume 2, pages 724–731, Monterey, CA, March 1995.
4. X. Yuan, D. Borup, J. W. Wiskin, M. Berggren, R. Eidens, and S. A. Johnson. Formulation and validation of Berenger's PML absorbing boundary for the FDTD simulation of acoustic scattering. *IEEE Trans. Ultrasonics, Ferroelectrics, Freq. Control*, 44(4):816–822, July 1997.
5. Q. H. Liu and J. Tao. The perfectly matched layer for acoustic waves in absorptive media. *J. Acoust. Soc. Am.*, 102(4):2072–2082, October 1997.
6. T. Xiao and Q. H. Liu. Finite difference computation of head-related transfer function for human hearing. *J. Acoust. Soc. Am.*, 113(5):2434–2445, May 2003.
7. A. Taflove. Review of the formulation and applications of the finite-difference time-domain method for numerical modeling of electromagnetic wave interactions with arbitrary structures. *Wave Motion*, 10(6):547–582, 1988.
8. K. L. Shlager and J. B. Schneider. Comparison of the dispersion properties of several low-dispersion finite-difference time-domain algorithms. *IEEE Trans. Antennas Propagat.*, 51(3):642–653, March 2003.
9. E. A. Forgy and W. C. Chew. An efficient FDTD algorithm with isotropic numerical dispersion on an overlapped lattice. In *IEEE Antennas and Propagat. Soc. Int. Symposium*, volume 4, pages 1812–1815, Atlanta, GA, June 1998.
10. E. A. Forgy. A time-domain method for computational electromagnetics with isotropic numerical dispersion on an overlapped lattice. Master's thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1998.
11. E. A. Forgy and W. C. Chew. A time-domain method with isotropic dispersion and increased stability on an overlapped lattice. *IEEE Trans. Antennas Propagat.*, 50(7):983–996, July 2002.
12. J. von Neumann and R. D. Richtmyer. On the numerical solutions of partial differential equations of parabolic type. In A. H. Taub, editor, *John von Neumann Collected Works*, volume V, pages 652–663. Pergamon Press, 1963. Los Alamos Report LA-657, December 25, 1947.
13. A. Taflove and S. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 2 ed. Artech House, Boston, MA, 2000.
14. J. W. Nehrass, J. O. Jevtić, and R. Lee. Reducing the phase error for finite-difference methods without increasing the order. *IEEE Trans. Antennas Propagat.*, 46(8):1194–1201, August 1998.
15. J. B. Schneider and C. L. Wagner. FDTD dispersion revisited: Faster-than-light propagation. *IEEE Microwave Guided Wave Lett.*, 9(2):54–56, February 1999.
16. C. L. Wagner and J. B. Schneider. On the analysis of resonators using finite-difference time-domain techniques. *IEEE Trans. Antennas Propagat.*, 51(10):2885–2890, October 2003.
17. J. B. Schneider and R. J. Kruhlak. Dispersion of homogeneous and inhomogeneous waves in the Yee finite-difference time-domain grid. *IEEE Trans. Microwave Theory Techniques*, 49(2):280–287, February 2001.
18. J. B. Schneider, C. L. Wagner, and R. J. Kruhlak. Simple conformal methods for finite-difference time-domain modeling of pressure-release surfaces. *J. Acoust. Soc. Am.*, 104(6):3219–3226, December 1998.
19. D. E. Merewether, R. Fisher, and F. W. Smith. On implementing a numeric Huygen's source scheme in a finite difference program to illuminate scattering bodies. *IEEE Trans. Nuclear Sci.*, 27(6):1829–1833, December 1980.
20. A. Taflove and K. Umashankar. Radar cross section of general three-dimensional scatterers. *IEEE Trans.*

20 *Wagner and Schneider*

Electromag. Compat., EMC-25(4):433–440, November 1983.

21. J. Fang. *Time Domain Finite Difference Computation for Maxwell's Equations*. PhD thesis, University of California at Berkeley, Berkeley, CA, 1989.
22. J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comp. Phys.*, 114(1):185–200, 1994.
23. W. C. Chew and W. H. Weedon. A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates. *Microwave Opt. Tech. Lett.*, 7(13):599–604, September 1994.
24. H. Medwin and C. S. Clay. *Fundamentals of Acoustical Oceanography*. Academic Press, New York, 1998.
25. S. Wolfram. *The Mathematica Book*. Cambridge University Press, Cambridge, UK, 4 edition, 1999.