THEORETICAL BASIS FOR NUMERICALLY EXACT THREE-DIMENSIONAL TIME-DOMAIN ALGORITHMS

By

CHRISTOPHER LINCOLN WAGNER

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

May 2004

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of CHRISTOPHER LINCOLN WAGNER find it satisfactory and recommend that it be accepted.

_____
Chair

_____

_____

_____

ACKNOWLEDGMENTS

This work has been supported in part by Harold and Diana Frank EECS graduate fellowship, private funds, and Office of Naval Research Code 321OA.

THEORETICAL BASIS FOR NUMERICALLY EXACT

THREE-DIMENSIONAL TIME-DOMAIN

ALGORITHMS

Abstract

by Christopher Lincoln Wagner, Ph.D.
Washington State University
May 2004

Chair: John Brand Schneider

Applied engineering often requires one to obtain solutions to partial differential equations. Simple problems can be solved by analytic methods, but most practical application problems are too complicated to solve analytically. One must either solve a simplified approximation, or use numerical computations. There are numerous approximations and numerical methods. In this work solution techniques are considered for the time evolution of acoustic and electromagnetic fields. A popular time evolution method is Yee's finite-difference time-domain algorithm. This algorithm is used here as a reference algorithm. Unfortunately, the Yee algorithm introduces non-physical algorithmic anisotropic dispersion error. Some of the consequences of the anisotropic behavior of the Yee algorithm are considered. Then the spatial-differential operators needed to obtain theoretically numerically exact propagation in three-dimensions, with the central-difference time-derivative, are defined, analyzed, and demonstrated. Numerically exact is taken to mean that, to within the sampling limit imposed by the discretization in space and time, the only errors are due to the finite precision of digital computer arithmetic.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEDICATION

This work is dedicated to the faculty, family, and friends

who helped make this work possible.

Chapter 1

Introduction

The classic Yee Finite-Difference Time-Domain (FDTD) algorithm employs standard central-differences to approximate both the spatial and temporal derivatives of the governing equations [1]. These central differences yield second-order accuracy in terms of propagation in a homogeneous grid [2]. 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 electromagnetics problems, Maxwell's coupled differential equations are approximated as coupled finite difference equations. The finite-difference equations are solved for the future fields in terms of known past fields. The coupled equations are then alternately solved to advance the fields in the time-domain. 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. Given unlimited computer resources, any desired model fidelity could be obtained. 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 [3]. This dissertation is divided into chapters, briefly described below.

In order to formalize and simplify the analysis of FDTD algorithms, a finite-difference calculus is presented in Chapter 2. This finite-difference calculus is then used to perform dispersion and stability analysis of the Yee algorithm. The same basic techniques are used for the dispersion and stability analysis in the subsequent chapters.

In Chapter 3, the consequences of the Yee algorithm anisotropic dispersion is studied. The problem of mode scrambling is introduced and explained. Because a resonator with perfect electrically conducting (PEC) walls has no complications with absorbing boundary conditions and, for simple canonical geometries, the resonant frequencies are trivial to find, resonators are often used for analyzing the performance of finite-difference time-domain (FDTD) methods. However, when testing the performance of boundary implementations in an FDTD scheme one should compare to the resonant frequencies of a "perfect" discretized resonator (not to the mode frequencies in the continuous world). On the other hand, when testing the dispersion properties of a method, the resonant frequencies for some structures can be obtained directly from the dispersion relation, thus obviating the need for any simulation. Here we demonstrate how the dispersion relation can be used to obtain all the resonant frequencies of a rectangular resonator modeled with the Yee algorithm. Furthermore it is shown that modes that are degenerate in the continuous world can split into distinct modes in FDTD resonators, while modes that are separate in the continuous world

can combine in FDTD resonators, thus yielding extra or missing modes. Analytic results are verified using numerical simulations.

In Chapter 4 the problem of anisotropic dispersion is examined in the scalar field acoustic problem. 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.

In Chapter 5 the theoretical basis for numerically exact algorithms is presented. In a one-dimensional (1D) homogeneous space the classic Yee finite-difference time-domain (FDTD) algorithm is numerically exact when operated at the Courant stability limit. Numerically exact is taken to mean that, to within the sampling limit imposed by the discretization in space and time, the only errors are due to the finite precision of digital computer arithmetic. Unfortunately, the Yee algorithm is not numerically exact in two or more dimensions. However, using the design shown in Chapter 5, three-dimensional (3D) spatial differential operators can have 1D dispersion properties. Just as the space and time errors can be made to cancel in the 1D Yee algorithm, 3D algorithms (for hyperbolic systems

of coupled first order equations) in an unbounded homogeneous space can be constructed which are, in theory, numerically exact. The volumetric differential operators presented here extend over a localized non-zero volume, unlike the usual Nabla (or Del) operator which acts at a point. The computer implementations of volume operators presented here are based on reconstruction methods. Thus producing global range operators, so the implementations of these operators are computationally expensive. Sample implementations of approximate acoustic and electromagnetic algorithms are described and are shown to produce results that are superior to the classic Yee algorithm for the cubic resonator problem.

Chapter 2

Analysis of the Classic FDTD Technique

2.1 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

$$D_n = \frac{S_n(\frac{1}{2}) - S_n(-\frac{1}{2})}{\Delta_n}, \quad n \in (x, y, z, t). \tag{2.1}$$

Averaging operators are also needed

$$A_n = \frac{S_n(1) + S_n(-1)}{2}, \quad n \in (x, y, z). \tag{2.2}$$

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, $P(x, y, z, t) = e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)}$, the

difference operators acting on the wave yield

$$\begin{aligned} D_n P &= \frac{2i}{\Delta_n} \sin\left(\frac{k_n \Delta_n}{2}\right) P, \quad n \in (x, y, z), \\ D_t P &= -\frac{2i}{\Delta_t} \sin\left(\frac{\omega \Delta_t}{2}\right) P, \end{aligned} \tag{2.3}$$

where k_n, is the component of the wavevector in the n direction. When acting on a plane wave the averaging operators yield

$$A_n P = \cos(k_n \Delta_n) P, \quad n \in (x, y, z). \tag{2.4}$$

In this notation the Yee Nabla or Del operator ($\boldsymbol{\nabla}_0$) is given by

$$\boldsymbol{\nabla}_0 = D_x \hat{\mathbf{x}} + D_y \hat{\mathbf{y}} + D_z \hat{\mathbf{z}}. \tag{2.5}$$

The Yee gradient of a scalar plane wave is then

$$\begin{aligned} \boldsymbol{\nabla}_0 P(x, y, z) &= (D_x \hat{\mathbf{x}} + D_y \hat{\mathbf{y}} + D_z \hat{\mathbf{z}}) P(x, y, z) \\ &= \left(\frac{2i}{\Delta_x} \sin\left(\frac{k_x \Delta_x}{2}\right) \hat{\mathbf{x}} + \frac{2i}{\Delta_y} \sin\left(\frac{k_y \Delta_y}{2}\right) \hat{\mathbf{y}} + \frac{2i}{\Delta_z} \sin\left(\frac{k_z \Delta_z}{2}\right) \hat{\mathbf{z}} \right) P(x, y, z). \end{aligned} \tag{2.6}$$

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

$$\begin{aligned} \boldsymbol{\nabla}_0 \cdot \mathbf{V} &= D_x V_x + D_y V_y + D_z V_z \\ &= \frac{2i}{\Delta_x} \sin\left(\frac{k_x \Delta_x}{2}\right) V_x + \frac{2i}{\Delta_y} \sin\left(\frac{k_y \Delta_y}{2}\right) V_y + \frac{2i}{\Delta_z} \sin\left(\frac{k_z \Delta_z}{2}\right) V_z, \end{aligned} \tag{2.7}$$

where $\mathbf{V}$ is a vector plane wave. The Yee curl of a vector $\mathbf{V}$ is

$$\boldsymbol{\nabla}_0 \times \mathbf{V} = (D_y V_z - D_z V_y)\hat{\mathbf{x}} + (D_z V_x - D_x V_z)\hat{\mathbf{y}} + (D_x V_y - D_y V_x)\hat{\mathbf{z}}, \qquad (2.8)$$

where for plane waves the terms can be expanded with (2.3).

2.2 Dispersion and Stability Analysis

A simple technique for analyzing the stability of finite difference methods was given by von Neumann and Richtmyer [4]. In summary, substituting a complete basis set of complex-exponential plane-wave solutions into the finite difference equations of the system converts time and space derivatives into multiplicative factors. Then the fields are eliminated from the resulting algebraic system to obtain the dispersion relation. The dispersion relation can be analyzed to obtain stability requirements. The stability conditions are obtained by examining the dispersion equation for the relationship between the wave vector $\mathbf{k}$ and angular frequency ω. When the relationship is complex, the result is an unstable algorithm [4] or superluminal propagation [5]. The material parameters are assumed constant in the region of analysis. For a 3D space, plane waves of arbitrary wave-vector and frequency form a complete basis. While von Neumann did obtain a dispersion relation (without calling it such), he only considered stability properties. Taflove and Brodwin [6] demonstrated the utility of analyzing the dispersion relation for understanding propagation behavior, i.e., the accuracy of the simulation. Thus the dispersion relation for a time-domain algorithm provides a powerful tool for analyzing the algorithm for both stability and accuracy.

Obtaining a dispersion relation for a Yee algorithm [6] is simple with the finite difference calculus. As an example, the acoustics dispersion relation is derived. For homogeneous-space small-signal acoustics the governing equations are

$$\frac{\partial \mathbf{V}}{\partial t} = -\frac{1}{\varrho}\boldsymbol{\nabla} P, \qquad \frac{\partial P}{\partial t} = -\varrho c^2 \boldsymbol{\nabla} \cdot \mathbf{V}, \tag{2.9}$$

where ϱ is the density, c is the wave speed, and $\mathbf{V}$ and P are the velocity and pressure fields, respectively. Applying (2.3), (2.6), and (2.7) to each term of (2.9) for assumed plane wave solutions with wave vector $\mathbf{k}$ and frequency ω yields

$$\begin{aligned} D_t \mathbf{V} &= -\frac{1}{\varrho}\boldsymbol{\nabla}_0 P = -\frac{1}{\varrho}(\hat{\mathbf{x}} D_x P + \hat{\mathbf{y}} D_y P + \hat{\mathbf{z}} D_z P), \\ D_t P &= -\varrho c^2 \boldsymbol{\nabla}_0 \cdot \mathbf{V} = -\varrho c^2 (D_x V_x + D_y V_y + D_z V_z). \end{aligned} \tag{2.10}$$

Solving (2.10) simultaneously to eliminate (for example) $\mathbf{V}$ yields

$$D_t^2 P = c^2 \boldsymbol{\nabla}_0 \cdot \boldsymbol{\nabla}_0 P. \tag{2.11}$$

Substituting the expressions for the plane-wave finite-differences (2.3), (2.6), and (2.7) results in the Yee dispersion relation [6]

$$\left(\frac{1}{c\Delta_t}\sin\frac{\omega\Delta_t}{2}\right)^2 = \left(\frac{1}{\Delta_x}\sin\frac{k_x\Delta_x}{2}\right)^2 + \left(\frac{1}{\Delta_y}\sin\frac{k_y\Delta_y}{2}\right)^2 + \left(\frac{1}{\Delta_z}\sin\frac{k_z\Delta_z}{2}\right)^2. \tag{2.12}$$

For cubic cells of step size Δ_g, this reduces to

$$\frac{1}{S^2}\sin^2\left(\frac{\omega\Delta_t}{2}\right) = \sin^2\left(\frac{k_x\Delta_g}{2}\right) + \sin^2\left(\frac{k_y\Delta_g}{2}\right) + \sin^2\left(\frac{k_z\Delta_g}{2}\right), \tag{2.13}$$

where S is the Courant number equal to $c\Delta_t/\Delta_g$.

To find the Courant stability limit the dispersion relation (2.13) is solved for the frequency

$$\omega = \frac{2}{\Delta_t}\sin^{-1}\left(S\left[\sin^2\left(\frac{k_x\Delta_g}{2}\right) + \sin^2\left(\frac{k_y\Delta_g}{2}\right) + \sin^2\left(\frac{k_z\Delta_g}{2}\right)\right]^{1/2}\right). \tag{2.14}$$

The Courant stability limit can be found using complex frequency analysis [4, 7] where stability requires that ω be real for all real k's permitted by grid sampling. Thus, in (2.14), the magnitude of the argument of the arcsine must be less than or equal to one. Stability limit S_L is given by

$$\frac{1}{S_L} = \max_{\substack{\mathbf{k} \\ |k_x|,|k_y|,|k_z| \le \frac{\pi}{\Delta_g}}} \left(\sqrt{\sin^2\left(\frac{k_x\Delta_g}{2}\right) + \sin^2\left(\frac{k_y\Delta_g}{2}\right) + \sin^2\left(\frac{k_z\Delta_g}{2}\right)}\right). \tag{2.15}$$

The maximum case will occur when all k_i's are π/Δ_g, so the stability limit is

$$S_L = \frac{1}{\sqrt{3}}. \tag{2.16}$$

The essential consequences of the dispersion relation can be understood with the help of figure 2.1, a graph of frequency ω vs. wavenumber $|k|$ in three grid directions, as given

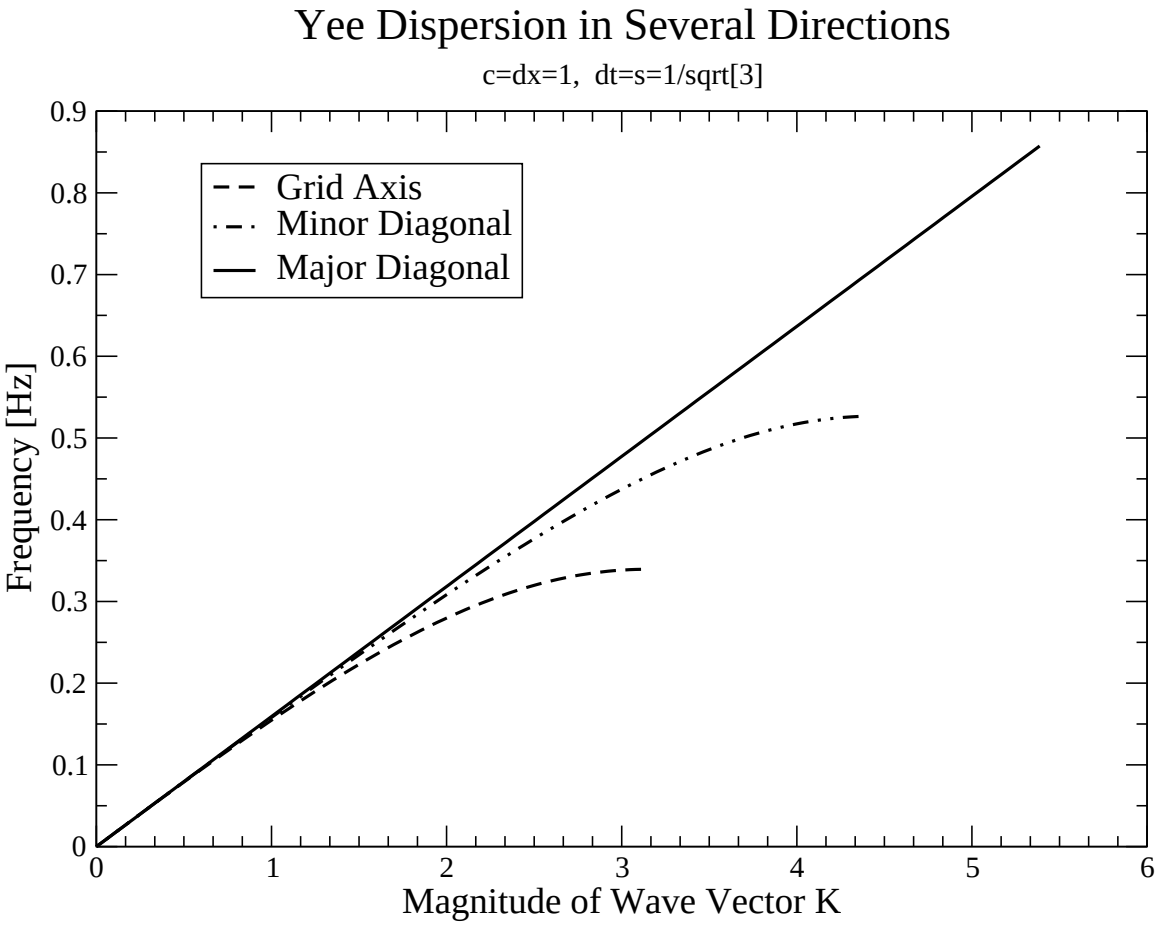

Figure 2.1: Yee algorithm dispersion curves ω vs. $|k|$ in three grid directions; the grid axis (1D), the planar diagonal (2D), and the major diagonal (3D). For fixed $|k|$ the anisotropic dispersion curves show a range of frequencies which will correspond to the single value of $|k|$. For fixed ω, the anisotropic dispersion will result in a range of $|k|$'s.

in (2.15). In resonator problems, the $|k|$ is fixed by the geometry and boundary conditions. For fixed $|k|$ the anisotropic dispersion curves show a range of frequencies which can correspond to the single value of $|k|$. For problems that fix the frequency ω (e.g., driving an antenna with a sinusoid), the anisotropic dispersion will result in a range of $|k|$'s for a single ω. The spread in $|k|$, and hence wavelength, is typically described as phase error.

Chapter 3

Understanding Numerical Dispersion in Electromagnetics FDTD

3.1 Introduction

Canonical resonators have been used to quantify the performance of FDTD methods designed to model boundaries that are not aligned with the grid (e.g., [8, 9]). Resonators or resonant-like structures have also been used in simulations to demonstrate the dispersion properties of a scheme (e.g., [10] where a parallel plate waveguide was used). When the goal is to ascertain the quality of the implementation of boundary conditions (such as the use of a locally conformal scheme to realize a PEC boundary), one should try to separate the errors introduced by inherent grid dispersion from those introduced by the boundary conditions themselves. To accomplish this, the resonant frequencies obtained in a simulation should be compared not to frequencies of the corresponding resonator in the continuous world, but rather to the frequencies of a "perfect" discretized scatterer, i.e., one that suffers the inherent anisotropy and dispersion of the grid, but does not have any boundary errors. We show that the dispersion relation for the Yee algorithm can be used to predict precisely the frequencies at which modes will oscillate in a rectangular resonator. Due to the

anisotropic dispersion of the Yee algorithm, modes that have degenerate frequencies in the continuous world may split into distinct frequencies in the discretized world. Conversely, other modes that one would anticipate are distinct may combine in an FDTD simulation. Given the ability to obtain the resonant behavior of some structures directly from the dispersion relation, it seems unnecessary to perform simulations using a resonator to quantify the dispersion properties of a given method. These simulations yield no insight into the method that is not already implicitly contained in the dispersion relation itself. However if the dispersion relation changes in the domain, for example at the boundary, then this simple dispersion analysis will not work. This is not a problem with the second order Yee algorithm, however higher order Yee, Forgy, or other algorithms may have different dispersion properties at boundaries.

We start by reviewing the Yee dispersion relation. We then show how it can be used to predict the resonances that will be present in a rectangular structure. Mode shifting, splitting, and combining are illustrated with simulations and with use of a frequency-versus-wavenumber diagram.

3.2 Dispersion Relation

The Yee dispersion relation, in rectangular coordinates, is given in (2.12). For cubic cells this reduces to (2.13), repeated here for convenience

$$\frac{1}{S^2}\sin^2\left(\frac{\omega\Delta_t}{2}\right) = \sin^2\left(\frac{k_x\Delta_g}{2}\right) + \sin^2\left(\frac{k_y\Delta_g}{2}\right) + \sin^2\left(\frac{k_z\Delta_g}{2}\right), \tag{3.1}$$

where S is the Courant number equal to $c\Delta_t/\Delta_g$, and Δ_g is the cell length. Note that the resonant wavenumbers are dictated by the physical size of the structure and this can be controlled precisely in the FDTD simulation. Thus the wavenumbers in the continuous and discrete worlds correspond exactly. However the frequencies that correspond to those wavenumbers differ.

3.3 Dispersed Resonator Frequencies

A rectangular resonator with PEC walls has resonant mode frequencies given by

$$\omega^2 = (\pi c)^2 \left[\left(\frac{m}{L_x} \right)^2 + \left(\frac{n}{L_y} \right)^2 + \left(\frac{p}{L_z} \right)^2 \right], \tag{3.2}$$

where m, n, and p are the mode indices, and L_x, L_y, and L_z are the size of the resonator in the x, y, and z directions respectively. In terms of the wavenumber components (3.2) can be written

$$\omega^2 = c^2 \left(k_x^2 + k_y^2 + k_z^2 \right). \tag{3.3}$$

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

$$k_x = \frac{m\pi}{L_x}, \qquad k_y = \frac{n\pi}{L_y}, \qquad k_z = \frac{p\pi}{L_z}. \tag{3.4}$$

Putting (3.1) and (3.4) together, and solving for $f = \omega/2\pi$, we obtain

$$f = \frac{1}{\pi\Delta_t} \arcsin \left(S \sqrt{ \left(\sin \frac{m\pi\Delta_g}{2L_x} \right)^2 + \left(\sin \frac{n\pi\Delta_g}{2L_y} \right)^2 + \left(\sin \frac{p\pi\Delta_g}{2L_z} \right)^2 } \right). \tag{3.5}$$

This gives the resonant frequency f for a particular set of mode indices for any Yee FDTD rectangular resonator. When a resonator has the same size in more than one dimension modes with permutations of indices will be degenerate. For example if $L_x = L_y$, modes $(1, 7, 0)$ and $(7, 1, 0)$ are degenerate. When a mode has this type of degeneracy we will refer to the mode in the plural even if a single set of indices is given.

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 $1/2\Delta_t$ [11]. In the continuum there are modes whose frequencies are below the grid Nyquist frequency, but that have wavenumber components that are complex [5, 11]. The transition between purely real and complex wavenumbers occurs where there are exactly two grid points per numeric wavelength. Complex wavenumber components experience exponential decay and hence the corresponding mode does not resonate. Therefore the continuum theory is applied here with the understanding that the wavenumbers do not extend beyond those that are real in the FDTD grid. From (3.5) it appears that one can use any value for the mode indices and still obtain a real result (assuming S is less than the stability limit of $1/\sqrt{3}$). This is true of this equation but that observation masks, to some extent, the true behavior of the grid and what is realizable. Regardless of the direction of propagation, there must be at least two samples per numeric wavelength, i.e., the minimum wavelength, λ_{min}, is $2\Delta_g$. The wavenumber corresponding to this discretization is $k_{\text{max}} = 2\pi/\lambda_{\text{min}} = \pi/\Delta_g$. Using this as a bound on a single wavenumber component, say the k_x component, and equating with the expression given in (3.4), we obtain $k_x = m\pi/L_x \leq \pi/\Delta_g$. This places a bound on the

mode index m such that $m \leq L_x/\Delta_g$. Therefore, if the resonator size L_x is N cells, then m can be no larger than N. Similar arguments hold in the other directions.

3.4 FDTD Simulations

Here we demonstrate the agreement of measured and predicted values for FDTD resonators. The rectangular resonators are excited by a single element current source in the z direction, centered in the domain. The source is a unit amplitude current of duration $2\Delta_t$, giving a spectral null at the time-stepping Nyquist frequency. For operation at the stability limit, the time-stepping Nyquist frequency will correspond to the vector $\mathbf{k}$ spatial-sampling frequency. The z component of the electric field is sampled at the location of the source. This data is Fourier transformed to produce the mode spectral plots shown below. With this geometry the excited and detectable modes will have odd x and y mode indices, while the z axis index will be even. To maintain the source at the center of the domain, there must be an even number of cells in the x and y directions and an odd number in the z direction. For simplicity, the domain is kept as nearly cubic as possible so the number of cells in the z direction is one less than in x and y. Unit cells are cubic, with E fields along the cell edges. Without loss of generality, unit cells are assigned a size of $\Delta_g = 1$ m. The simulations use the Courant limit ($1/\sqrt{3}$) and are run for 65536 time steps. An FFT was used to obtain spectral information. Using these parameters the Nyquist frequency is 259.6278 MHz and the spectral resolution Δf is 7.923211 kHz. In the spectral plots to follow, symbols are used to designate continuum and FDTD predicted resonant frequencies while a line shows the result of the FDTD simulation (i.e., a solid line shows the entire spectrum

obtained from the simulation). Specifically, plus signs are used to indicate the continuum frequencies obtained via (3.2) while X's are used to indicate FDTD frequencies obtained via (3.5).

The difference between the continuum resonance and the FDTD resonance, as well as the ability to predict the FDTD resonant frequency, can be demonstrated with a trivially small resonator. The smallest possible resonator with the geometry described above has a volume of $2 \times 2 \times 1$ cells. Such a resonator has only one mode, the $(1, 1, 0)$ mode. The continuum frequency is 105.9927 MHz. Equation (3.5) predicts the FDTD frequency will be 101.7291 MHz, this agrees within $\pm\Delta f$ of the observed FDTD results, as shown in Figure 3.1.

To illustrate more clearly nonlinear effects, a resonator with a size of $4 \times 4 \times 3$ is considered next. The amplitude spectrum is shown in Figure 3.2. Note that the FDTD predicted values match the observed spectral peaks whereas the continuum frequencies do not. The $(1, 3, 0)$ modes are dispersion shifted to a frequency lower than the $(1, 1, 2)$ mode, and the $(3, 3, 0)$ mode is shifted to below the $(1, 3, 2)$ modes as shown in Table 3.1. There is a large dc line [12] due to the charge deposited by the pulsed source. A dc line does not exist for the previous resonator because that small resonator did not provide ample room to store charge to either side of the source (i.e., the charge was effectively shorted by the walls of the resonator).

In Figure 3.2 note the relatively good agreement between the FDTD resonance and the continuum theory at the highest resonant frequency. This is, at first, somewhat counter-intuitive since this resonance occurs in a region with coarse discretization where one might

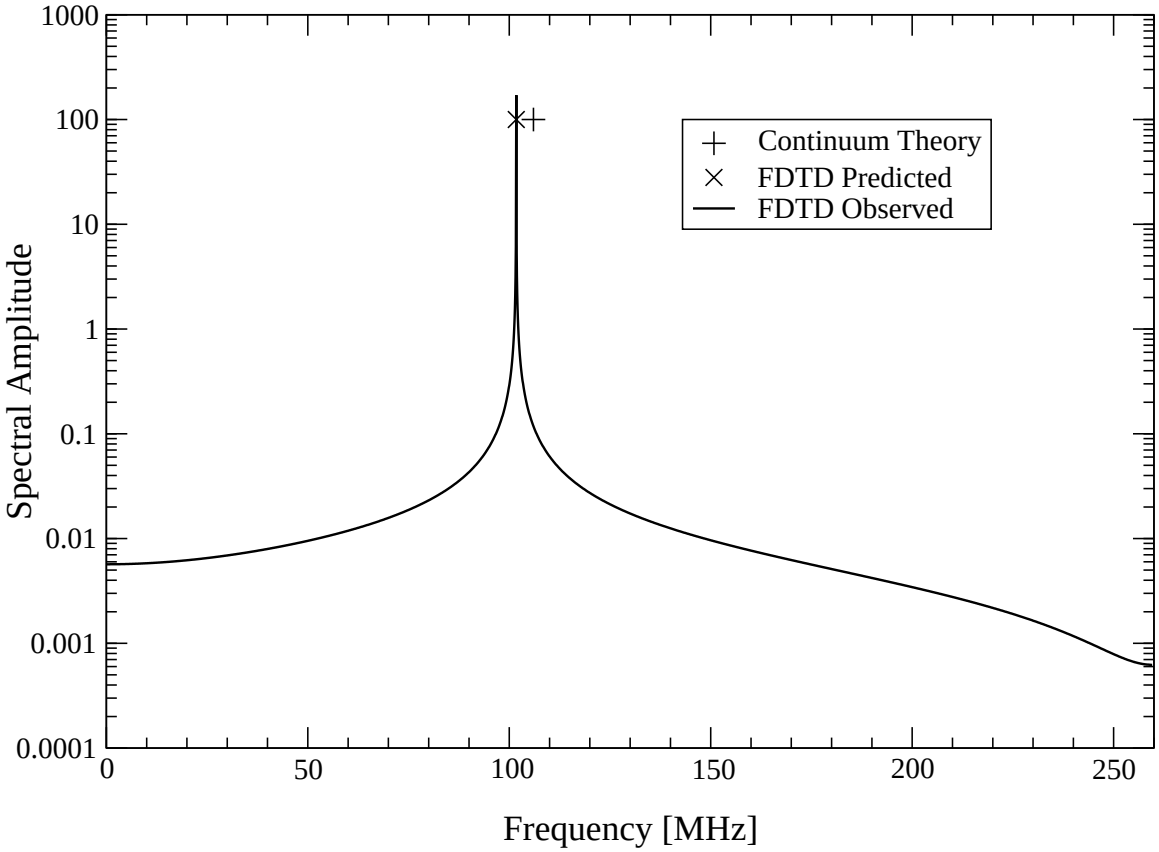

Figure 3.1: Continuum and predicted resonant frequencies as well as observed spectrum for a $2 \times 2 \times 1$ resonator. Continuum and FDTD predicted frequencies are indicated with symbols drawn with arbitrary amplitudes.

Mode	Continuum Theory	FDTD Predicted	FDTD Observed
$(1,1,0)$	52.99633	52.52425	52.52297
$(1,1,2)$	113.1140	104.2227	104.2219
$(1,3,0)$	118.5034	101.7291	101.7261
$(1,3,2)$	155.0136	143.6522	143.6557
$(3,3,0)$	158.9890	141.2610	141.2629
$(3,3,2)$	187.7862	187.0015	187.0036

Table 3.1: $4 \times 4 \times 3$ resonator frequencies. The observed frequencies are within $\pm\Delta f$ of the predicted values. All frequencies are in MHz.

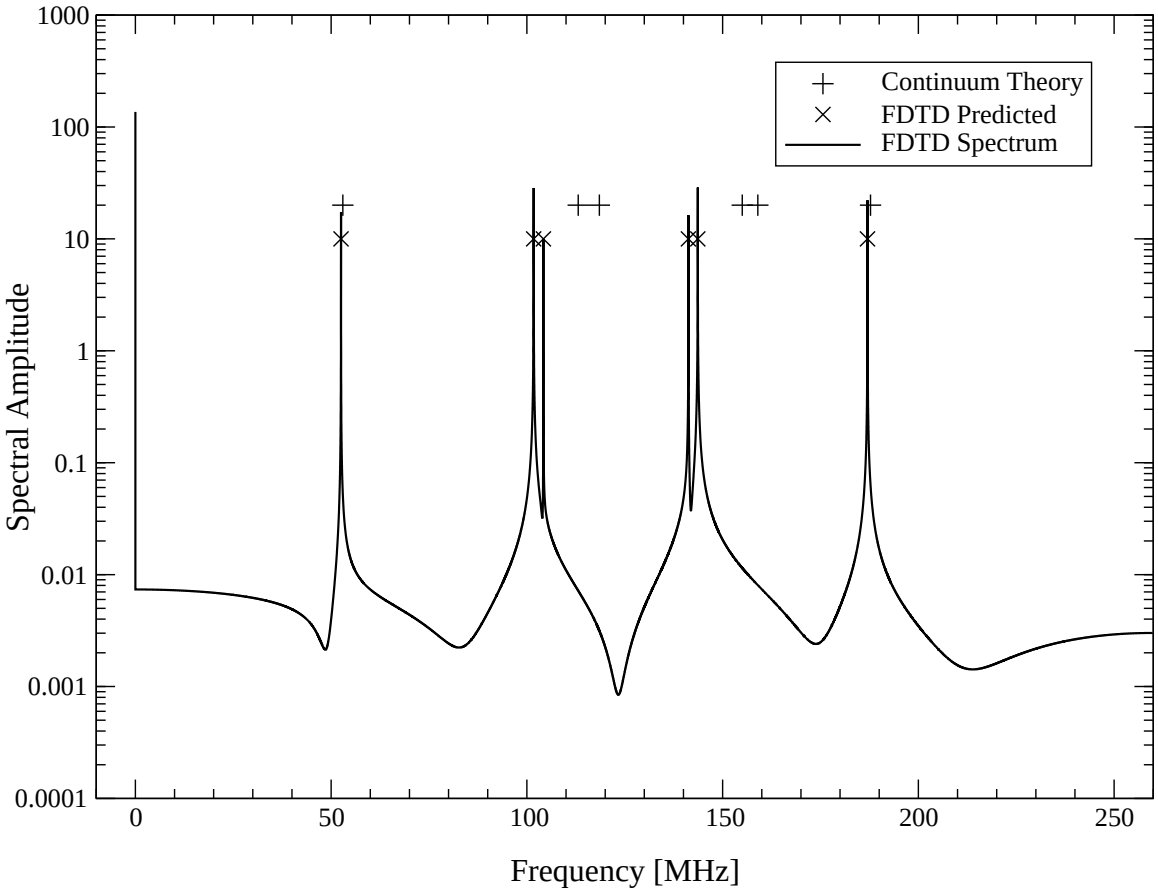

Figure 3.2: Continuum and FDTD predicted resonant frequencies as well as observed spectrum for a $4 \times 4 \times 3$ resonator. Note the good performance of the FDTD for the highest frequency mode. This mode is $(3, 3, 2)$ which is almost along the major diagonal. Continuum and FDTD predicted frequencies are plotted with arbitrary amplitude.

expect the worst dispersion errors. However, this peak corresponds to the $(3, 3, 2)$ mode whose associated wavevectors are nearly aligned with the grid diagonals. Since these simulations are run at the Courant limit there is no grid dispersion along the grid diagonals regardless of the discretization. This illustrates that, due to the anisotropy of the grid, the amount of dispersion is a function of both the discretization and the direction of propagation. For a general resonator in which one does not know the orientation of the associated wavevectors, one would be unable to say if the agreement between the FDTD-generated resonances and the continuum resonances are getting better or worse as the frequency increases. Nevertheless, one can be confident that agreement is good for well-resolved frequencies (i.e., ones for which the discretization is high enough to ensure low dispersion for all directions of propagation).

Mode splitting is a result of the anisotropic dispersion of the Yee grid. Modes with the same continuum frequency can have different dispersion shifts owing to their different wavenumber components. This occurs because continuum degenerate modes having distinct sets of indices suffer different dispersion in the FDTD grid thus resulting in two (or more) spectral lines. The smallest resonator where splitting occurs is the $8 \times 8 \times 7$ resonator. For example the $(1, 7, 0)$ modes, and the $(5, 5, 0)$ mode are degenerate. However, the dispersion shift will be different for the $(1, 7, 0)$ modes and $(5, 5, 0)$ mode, thus splitting this line. The same thing occurs if the third index is 2, 4 or 6. The measured and predicted resonances for the $8 \times 8 \times 7$ resonator are shown in Figure 3.3. Again, there is perfect agreement between the predicted and measured FDTD frequencies and these frequencies may differ substantially from the continuum frequency.

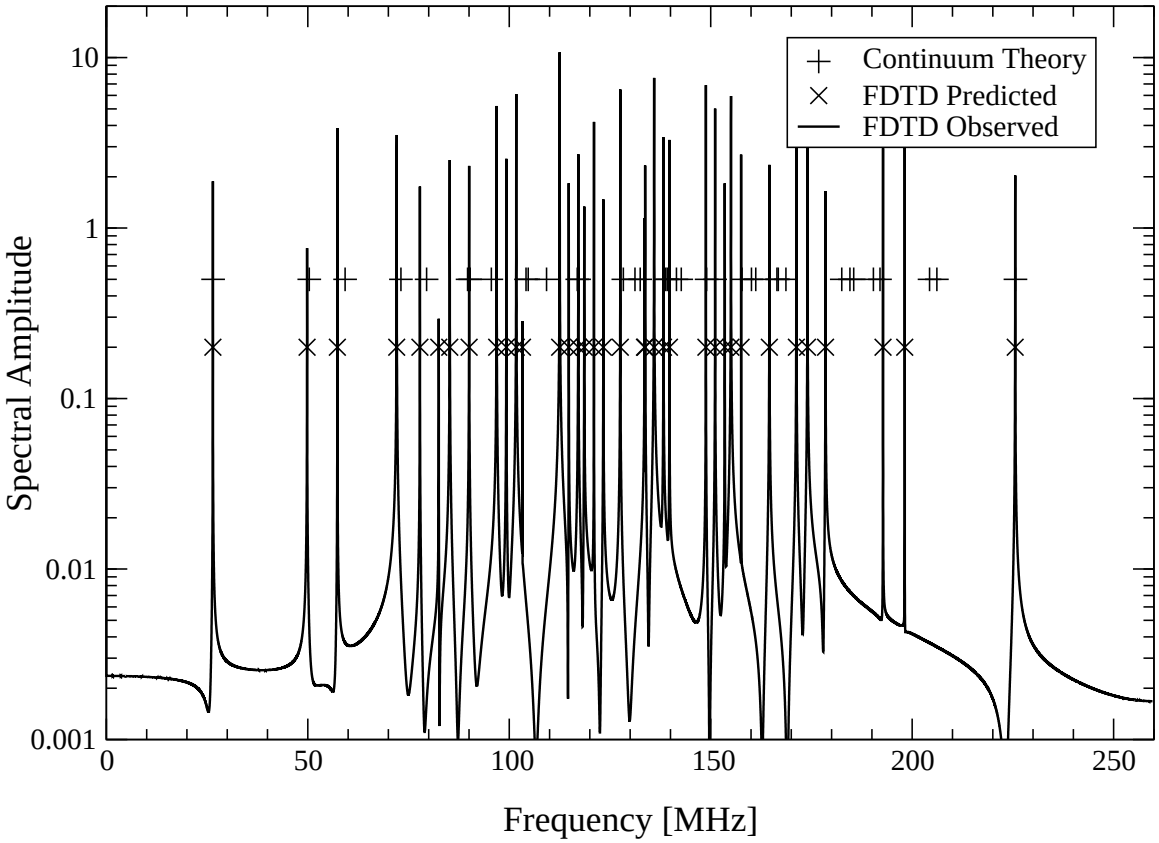

Figure 3.3: Continuum and FDTD predicted resonant frequencies as well as observed spectrum for a $8 \times 8 \times 7$ resonator. Continuum and FDTD predicted frequencies are plotted with arbitrary amplitude. The dc line is not shown.

The resonators illustrated here do not exhibit an extra total number of mode lines since some modes also combine while others split. However, once the size of the resonator is above $20 \times 20 \times 19$, the FDTD resonator will have more mode lines than the continuum theory predicts.

Mode combining occurs when two (or more) distinct modes suffer different dispersion resulting in the mode-lines combining to form a single line. In the case of the $8 \times 8 \times 7$ resonator, the $(1, 7, 0)$ modes are dispersion shifted to the same frequency as the dispersion-shifted $(3, 5, 0)$ mode, thus yielding a single spectral line in the simulation.

The mechanics of mode splitting and combining is perhaps best understood using an f-k (ω-β) diagram of the frequency versus the wavenumber. Figure 3.4 shows such a diagram that can be used to demonstrate the splitting and combining of the modes mentioned above. First, one draws a straight line of slope $c/2\pi$ representing the relationship between frequency and wavenumber magnitude in the continuum. Assume one is interested in the $(1, 7, 0)$ and $(5, 5, 0)$ modes which are degenerate in the continuum. The dispersion curves for these modes are added to the plot. To generate these curves, the magnitude of the wavenumber is now treated as the independent variable. For the $(1, 7, 0)$ mode, one plots

$$\frac{1}{S^2}\sin^2\left(\frac{2\pi f\Delta_t}{2}\right) = \sin^2\left(\frac{k\Delta_g}{2\sqrt{50}}\right) + \sin^2\left(\frac{7k\Delta_g}{2\sqrt{50}}\right), \tag{3.6}$$

while for the $(5, 5, 0)$ mode one plots

$$\frac{1}{S^2}\sin^2\left(\frac{2\pi f\Delta_t}{2}\right) = 2\sin^2\left(\frac{5k\Delta_g}{2\sqrt{50}}\right). \tag{3.7}$$

Now (3.2) is used to obtain the continuum frequency for the resonant mode. This point is identified in Figure 3.4 by the intersection of the straight continuum line and the horizontal line labeled "Continuum $(1, 7, 0)$ & $(5, 5, 0)$". Since the FDTD resonator must have the same wavenumbers, one draws a vertical line from that point on the continuum line and finds the intersections with the dispersion curves for the $(1, 7, 0)$ and $(5, 5, 0)$ modes. These intersections are indicated with the horizontal lines labeled "Observed (5,5,0)" and "Observed (1,7,0)". The vertical distance between these horizontal lines shows the difference in frequency of these supposedly degenerate modes.

Mode combining is illustrated in a similar fashion. One has to add the dispersion curve for the $(3, 5, 0)$ mode and identify the frequency for the corresponding continuum resonance. Drawing a vertical line from that point on the continuum line, one finds that the intersection with the $(3, 5, 0)$ dispersion curve is precisely at the same frequency as the $(1, 7, 0)$ mode. Hence these distinct modes in the continuum yield a single resonance in the FDTD simulation. The frequencies associated with these modes are given in Table 3.2.

3.5 Chapter Summary

The dispersion relation accurately predicts the frequencies at which a rectangular resonator mode will resonate. Dispersion can split or combine modes. Furthermore, the dispersion shift can change the resonant frequencies so that a list of observed modes ordered by resonant frequency may or may not correspond in order to a list obtained from the continuum.

The Yee dispersion anisotropy is responsible for mode splitting and combining. Al-

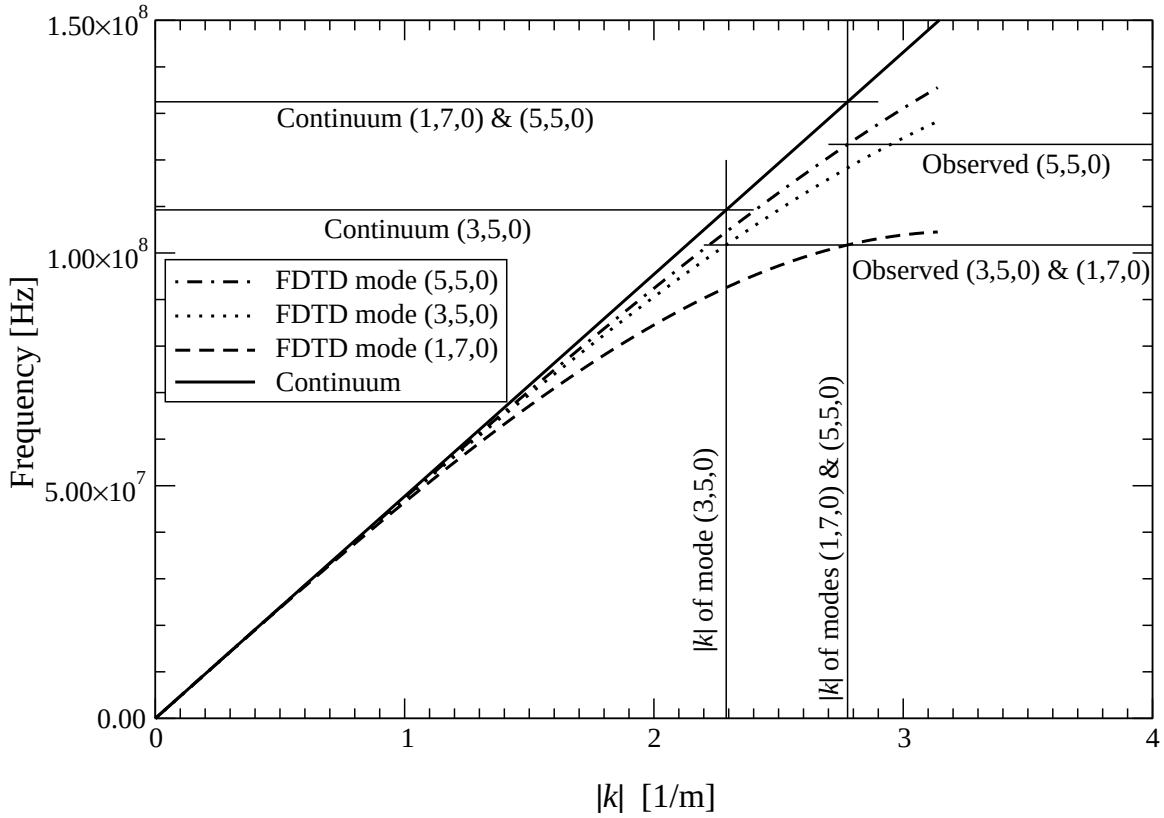

Figure 3.4: Dispersion curves (frequency vs. wavenumber) for selected modes in the $8 \times 8 \times 7$ resonator, graphically illustrating mode splitting and combining. Horizontal lines labeled "Continuum" are the frequencies which should exist for a given k. Lines labeled "Observed" are the dispersion shifted frequencies that will be seen in the FDTD simulation.

Mode	Continuum Theory	FDTD Predicted	FDTD Observed
$(3,5,0)$	109.2547	101.7291	101.7261
$(3,3,4)$	116.8595	114.7771	114.7756
$(3,5,2)$	117.3490	112.5123	112.5096
$(1,5,4)$	128.3151	121.0187	121.0191
$(1,1,6)$	131.1865	103.2788	103.2791
$(1,7,0)$	132.4908	101.7291	101.7261
$(5,5,0)$	132.4908	123.3438	123.3406

Table 3.2: Selected list of $8 \times 8 \times 7$ resonator mode frequencies, showing the effect of combining and splitting. The $(1,7,0)$ modes splits from the $(5,5,0)$ mode and combines with the $(3,5,0)$ modes, jumping over intervening modes. Similar splitting/combining occurs with a third index of 2, 4 or 6. Observed degenerate modes cannot be distinguished in the FDTD data, the observed line frequency closest to the predicted line is reported. All frequencies are in MHz.

gorithms which are more isotropic than Yee, such as the algorithm developed in the next chapter (based on the Forgy isotropic scheme [13] which is isotropic to fourth order), will have reduced mode splitting. Given the dispersion relation one can post-process resonance data to correct for the mean dispersion error. Such a correction would shift peaks but would not undo any splitting or combining.

The resonant analysis conducted here was limited to the Yee scheme, but, as demonstrated in the next chapter, a similar analysis can be conducted for any FDTD scheme for which a dispersion relation exists. It is to be noted that the rectangular resonators considered here have simple resonances in which the fields can be viewed as the superposition of plane waves propagating in discrete directions. Knowing the directions and the corresponding frequencies, and given the dispersion relation, one can exactly predict the amount of error in the resonant frequency obtained from a simulation. However, for a general resonator in which the wavevectors associated with a given mode are not known one cannot determine the amount of error, nor can one completely correct for that error, unless the particular scheme has the same amount of error for all wavevectors at a given discretization. The Yee scheme, which is anisotropic to second order, permits only limited correction where one can correct for the mean dispersion error. Such a technique is essentially the one applied in the dispersion correction technique described by Nehrbass, Jevtić, and Lee [14] (however they applied the correction at a single frequency prior to the simulation rather than as a post-processing correction across all frequencies). On the other hand, correcting for the mean dispersion error in the Forgy scheme (which is, same as the Yee algorithm, second-order accurate but, unlike the Yee algorithm, isotropic to fourth order) would yield

results that are more accurate than those which could be obtained from the (corrected) Yee algorithm at the same discretization.

Chapter 4

An Acoustics FDTD Algorithm with Improved Isotropy

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.

4.1 Introduction

Here the electromagnetic algorithm proposed by Forgy [15, 16, 13] 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. The algorithm is defined in Section 4.2 and

the dispersion relation is obtained in Section 4.3. The algorithm is shown to have second-order accuracy but fourth-order isotropy. Despite the fact that the algorithm has the same accuracy as the Yee algorithm, the 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 4.4. Another type of correction made possible by the increased isotropy is discussed in Section 4.5 where it is demonstrated that post-processing of homogeneous-domain spectra can correct much of the dispersion error over the entire spectrum. Section 4.6 presents simulations of cubic resonators. Finally, Section 4.7 presents simulations of plane-wave scattering from penetrable spheres. Results at multiple frequencies are compared to the exact solution.

4.2 The Constructed Divergence

The constructed 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 is defined by

$$\alpha_0 \boldsymbol{\nabla}_0 + \alpha_1 \boldsymbol{\nabla}_1, \tag{4.1}$$

where the α's are the combination weights (α_0 is the Yee weight), $\boldsymbol{\nabla}_0$ is given in (2.5), and $\boldsymbol{\nabla}_1$ is

$$\boldsymbol{\nabla}_1 = D_x \left(\frac{A_y + A_z}{2} \right) \hat{\mathbf{x}} + D_y \left(\frac{A_x + A_z}{2} \right) \hat{\mathbf{y}} + D_z \left(\frac{A_x + A_y}{2} \right) \hat{\mathbf{z}}. \tag{4.2}$$

One component of the $\boldsymbol{\nabla}_1$ operator averages four nodes, then takes the finite difference of the average of the opposite face, for example

$$\begin{aligned} \boldsymbol{\nabla}_1 \cdot \hat{\mathbf{x}} &= D_x \frac{A_y + A_z}{2} \\ &= \frac{(S_x(\frac{1}{2}) - S_x(-\frac{1}{2}))(S_y(1) + S_y(-1) + S_z(1) + S_z(-1))}{4\Delta_x}, \end{aligned} \tag{4.3}$$

The geometry implicit in (4.3) is shown in Figure 4.1. If $\alpha_0 = 1$ and $\alpha_1 = 0$ in (4.1), 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 Figure 4.2 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.

4.3 Dispersion Relation and Stability Limit

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

$$\frac{\partial \mathbf{V}}{\partial t} = -\frac{1}{\rho} \boldsymbol{\nabla} P, \tag{4.4}$$

$$\frac{\partial P}{\partial t} = -\rho c^2 \boldsymbol{\nabla} \cdot \mathbf{V} + P_{\text{source}}, \tag{4.5}$$

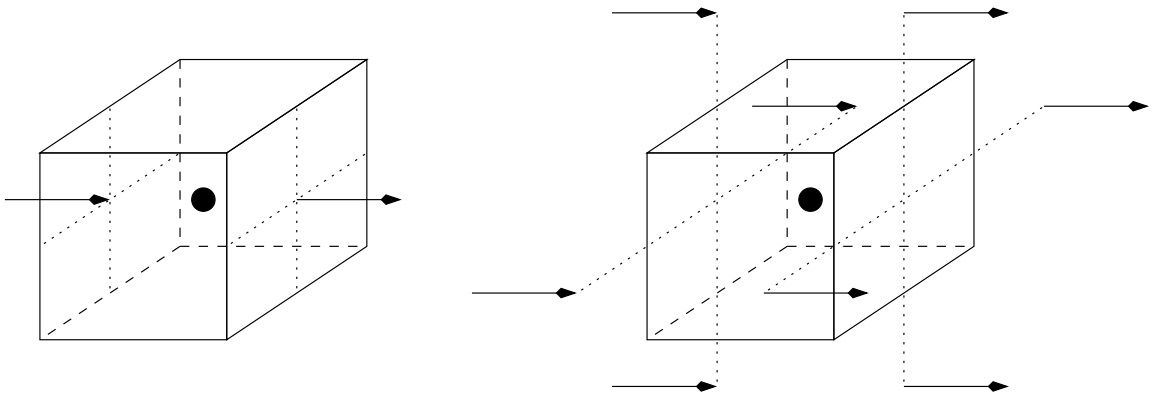

Figure 4.1: The diagram on the left shows the two velocity nodes which contribute to the x component of the Yee divergence, $\boldsymbol{\nabla}_0 \cdot \mathbf{V}$. The diagram on the right shows the eight velocity nodes which contribute to the x component of the $\boldsymbol{\nabla}_1 \cdot \mathbf{V}$ divergence.

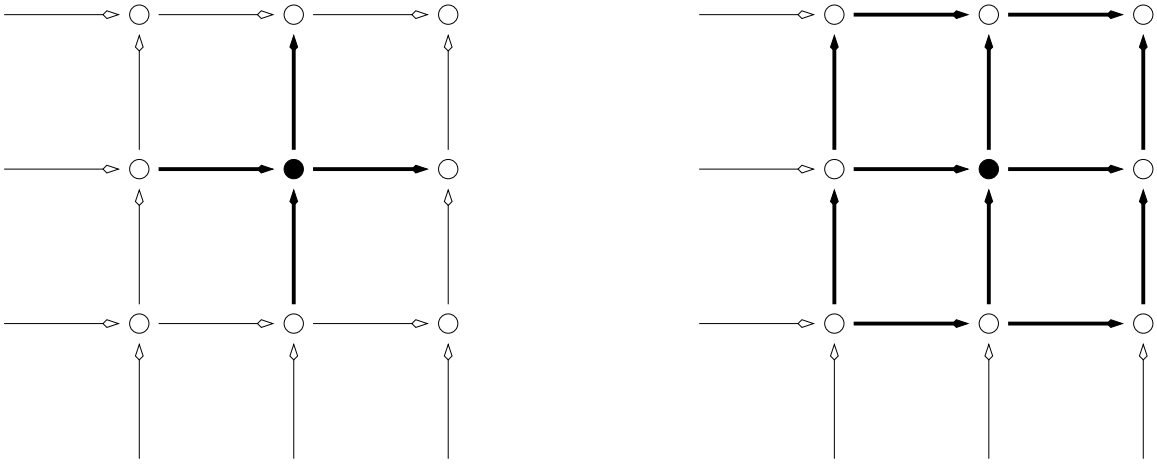

Figure 4.2: Two dimensional slice of the stencil of the Yee (left) and constructed divergence operators (right), acting in the upper right corner of a domain. The hollow circles are pressure nodes, the solid circle is the pressure node being updated by the divergence operator. The thin arrows are velocity nodes, and the bold arrows are velocity nodes contributing to the divergence operator.

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 (4.5) is different from the Yee FDTD update. To include the constructed operator's extra nodes (4.5) is written with (4.1) for the divergence operator

$$D_t P = -\rho c^2(\alpha_0 \boldsymbol{\nabla}_0 \cdot \mathbf{V} + \alpha_1 \boldsymbol{\nabla}_1 \cdot \mathbf{V}) + P_{\text{source}}. \tag{4.6}$$

Substituting the Yee implementation of (4.4) into the time derivative of (4.6) and assuming a source-free region produces the wave equation for the algorithm

$$D_t^2 P = c^2(\alpha_0 \boldsymbol{\nabla}_0 \cdot \boldsymbol{\nabla}_0 + \alpha_1 \boldsymbol{\nabla}_1 \cdot \boldsymbol{\nabla}_0) P. \tag{4.7}$$

Substituting the values of the operators from (2.3) and (2.4) into (4.7) and simplifying, produces the dispersion relation for the algorithm

$$\begin{aligned}\left(\frac{\Delta_g}{c\Delta_t}\right)^2 \sin^2\left(\frac{\omega\Delta_t}{2}\right) = &\sin^2\left(\frac{k_x\Delta_g}{2}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\big(\cos(k_y\Delta_g) + \cos(k_z\Delta_g)\big)\right) \\ &+ \sin^2\left(\frac{k_y\Delta_g}{2}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\big(\cos(k_x\Delta_g) + \cos(k_z\Delta_g)\big)\right) \\ &+ \sin^2\left(\frac{k_z\Delta_g}{2}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\big(\cos(k_x\Delta_g) + \cos(k_y\Delta_g)\big)\right),\end{aligned} \tag{4.8}$$

where the Δ_g 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 (4.8) 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 4.6, the modes dictate that the wave numbers for a supported mode are exact, which are then used in (4.8) 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 (4.8) in a power series in the space step size Δ_g, gives

$$\begin{aligned}(\alpha_0+\alpha_1)&(k_x^2+k_y^2+k_z^2)\\ &-\frac{1}{12}\left(\alpha_0(k_x^4+k_y^4+k_z^4)+\alpha_1(k_x^4+k_y^4+k_z^4+6k_x^2k_y^2+6k_x^2k_z^2+6k_y^2k_z^2)\right)\Delta_g^2\\ &+O[\Delta_g]^4.\end{aligned} \tag{4.9}$$

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

$$(k_x^2+k_y^2+k_z^2)-\frac{1}{12}(k_x^4+6\alpha_1k_y^2k_z^2+k_y^4+6\alpha_1k_x^2k_z^2+k_z^4+6\alpha_1k_x^2k_y^2)\Delta_g^2+O[\Delta_g]^4. \tag{4.10}$$

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

$$k^2-\frac{k^4\Delta_g^2}{12}+O[\Delta_g]^4. \tag{4.11}$$

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 dependant 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 (4.8) is solved for the frequency

$$\omega = \frac{2}{\Delta_t}\sin^{-1}\Bigg(S\Bigg[\sin^2\left(\frac{k_x\Delta_g}{2}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\big(\cos\left(k_y\Delta_g\right) + \cos\left(k_z\Delta_g\right)\big)\right) + \sin^2\left(\frac{k_y\Delta_g}{2}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\big(\cos\left(k_x\Delta_g\right) + \cos\left(k_z\Delta_g\right)\big)\right) + \sin^2\left(\frac{k_z\Delta_g}{2}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\big(\cos\left(k_x\Delta_g\right) + \cos\left(k_y\Delta_g\right)\big)\right)\Bigg]^{1/2}\Bigg), \tag{4.12}$$

where $S = c\Delta_t/\Delta_g$ is the Courant number. The Courant stability limit can be found using complex frequency analysis [4, 7] where stability requires that ω be real for all real k's permitted by grid sampling. Thus, in (4.12), 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_g. For $\alpha_1 \leq 1/4$ the limiting case is when $k_x = k_y = k_z = \pi/\Delta_g$, and when $1/4 < \alpha_1 \leq 1/2$ the limiting case is when two k's are π/Δ_g and one is zero. Using these

conditions the Courant limit is as follows

$$S = \begin{cases} \frac{1}{\sqrt{3}} & \text{for } \alpha_1 = 0, \text{ the Yee algorithm,} \\ \frac{1}{\sqrt{3(1-2\alpha_1)}} & \text{for } 0 \leq \alpha_1 \leq \frac{1}{4}, \\ \frac{1}{\sqrt{2(1-\alpha_1)}} & \text{for } \frac{1}{4} \leq \alpha_1 \leq \frac{1}{2}, \\ \frac{\sqrt{3}}{2} & \text{for } \alpha_1 = \frac{1}{3}, \text{ the present isotropic algorithm,} \\ 1 & \text{for } \alpha_1 = \frac{1}{2}, \text{ maximum } \alpha_1 \text{ for stable algorithm.} \end{cases} \tag{4.13}$$

To illustrate the effect of the constructed operator on the FDTD algorithm Figures 4.3 and 4.4 show the phase velocity error surfaces for the Yee and the present algorithm. Both figures are plots of $1 - \tilde{\omega}/\omega$ at $k = 2\pi/(10\Delta_g)$ (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.

4.4 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 [16]. The spatial derivatives are optimized with the same procedure as Forgy uses, however the time optimization is done here using the dispersion relation to prevent the Courant number shift of [16]. Optimization at a particular frequency involves

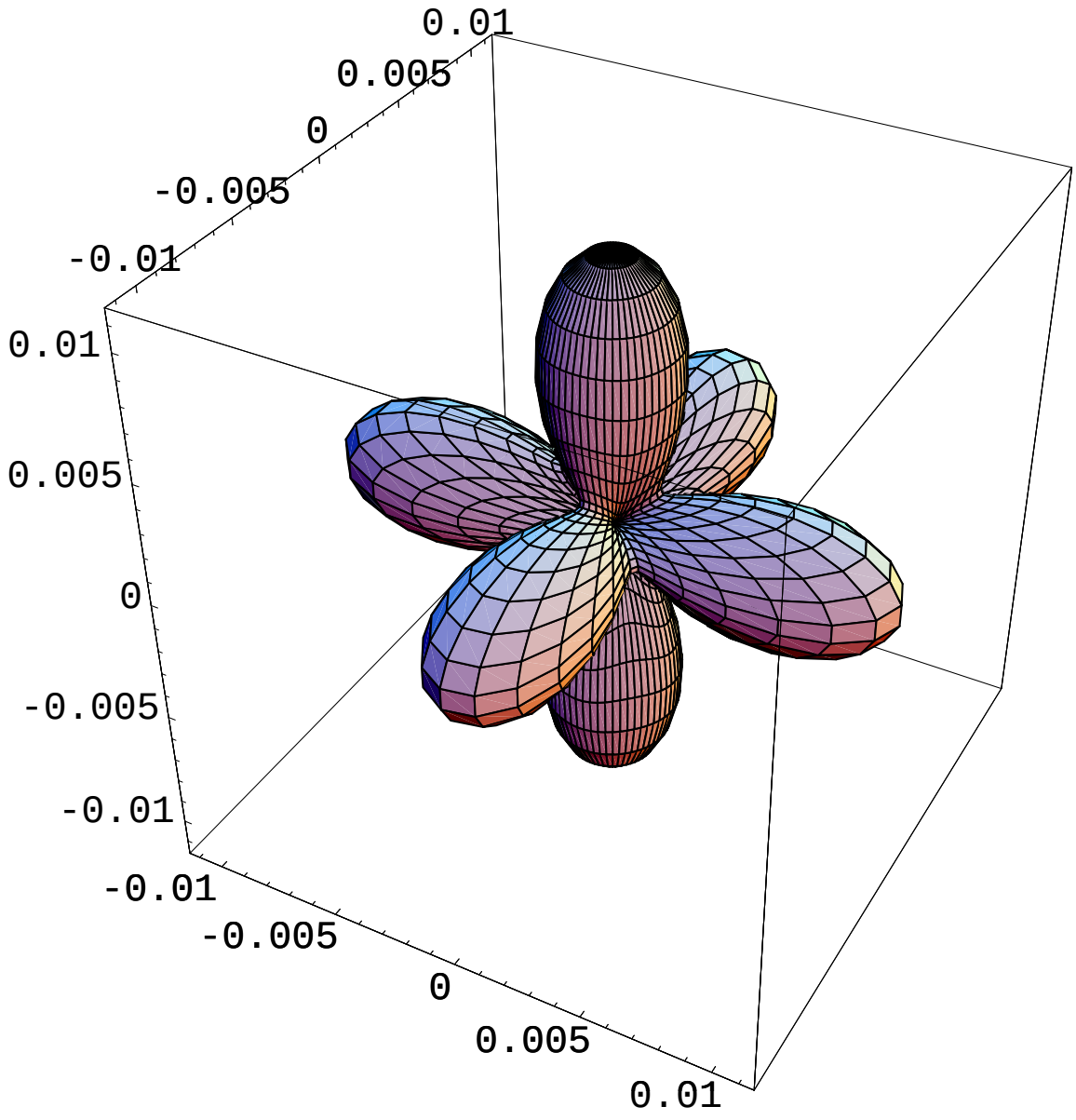

Figure 4.3: Dispersion error surface for the Yee algorithm at 10 samples per wavelength. The Courant number is $1/\sqrt{3}$ and $\alpha_1 = 0$.

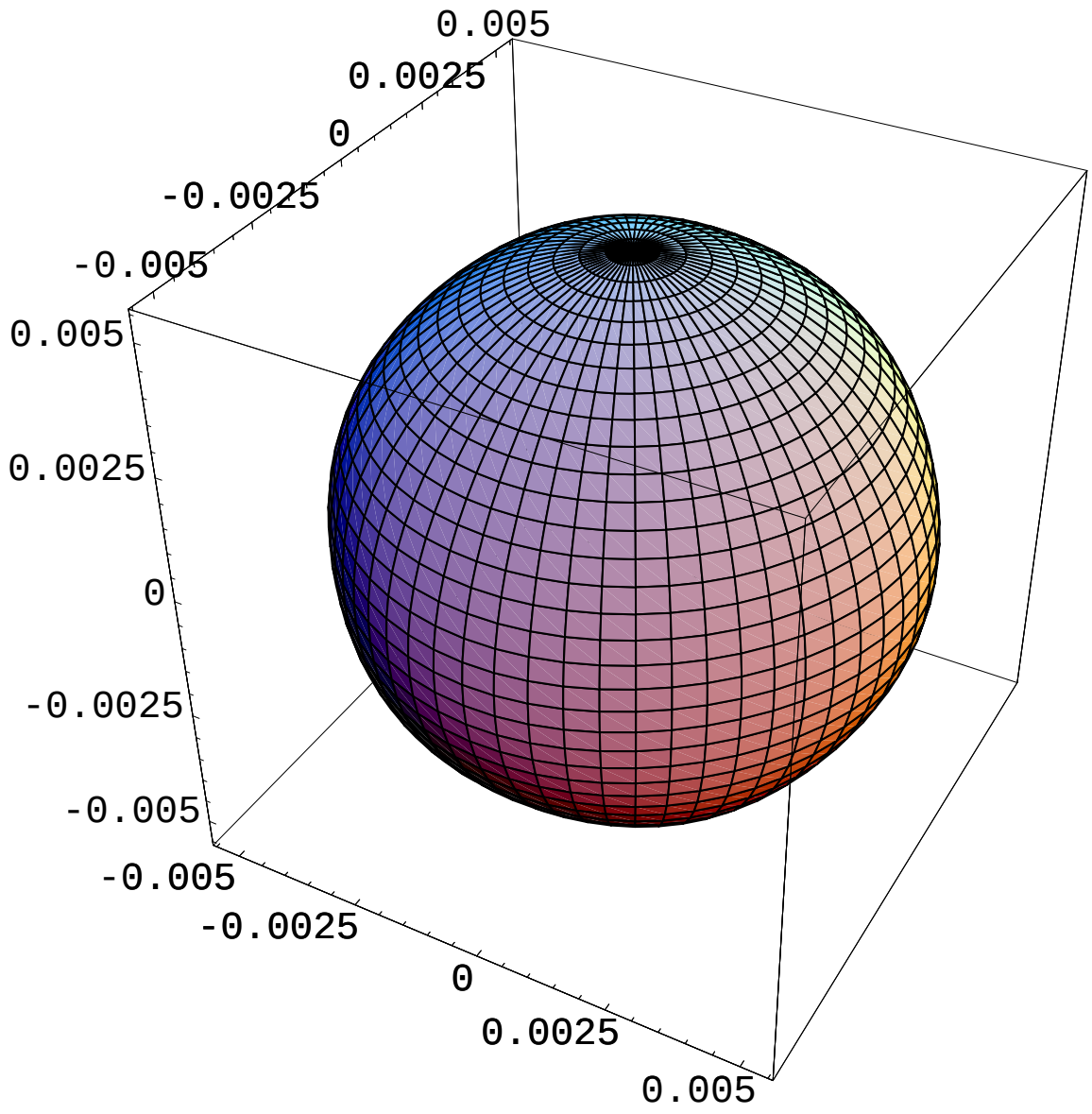

Figure 4.4: Dispersion error surface for the present algorithm 10 samples per wavelength. Note the smaller scale factors compared to Figure 4.3. The Courant number is $\sqrt{3}/2$ and $\alpha_1 = 1/3$.

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 (4.13).

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

$$\left(\frac{2}{\Delta_g}\right)^2 (\alpha_0 + \alpha_1) \sin^2\left(\frac{k_{\text{opt}}\Delta_g}{2}\right), \tag{4.14}$$

with the spatial dispersion along the major diagonal

$$\frac{12}{\Delta_g^2}\left[\alpha_0 + \alpha_1 \cos\left(\frac{k_{\text{opt}}\Delta_g}{\sqrt{3}}\right)\right] \sin^2\left(\frac{k_{\text{opt}}\Delta_g}{2\sqrt{3}}\right), \tag{4.15}$$

which produces

$$(\alpha_0 + \alpha_1) \sin^2\left(\frac{k_{\text{opt}}\Delta_g}{2}\right) = 3\left[\alpha_0 + \alpha_1 \cos\left(\frac{k_{\text{opt}}\Delta_g}{\sqrt{3}}\right)\right] \sin^2\left(\frac{k_{\text{opt}}\Delta_g}{2\sqrt{3}}\right). \tag{4.16}$$

The spatial optimization is performed by simultaneously solving (4.16) and $\alpha_0 + \alpha_1 = 1$ for α_0 and α_1 giving

$$\alpha_0 = (1 - \alpha'), \quad \alpha_1 = \alpha', \tag{4.17}$$

where

$$\alpha' = \frac{2 + \cos\left(k_{\text{opt}}\Delta_g\right) - 3\cos\left(\frac{k_{\text{opt}}\Delta_g}{\sqrt{3}}\right)}{12\sin^4\left(\frac{k_{\text{opt}}\Delta_g}{2\sqrt{3}}\right)}. \tag{4.18}$$

The optimization to this point ensures that the dispersion-shifted frequencies $\tilde{\omega}$ found via (4.12) are equal for propagation along the grid axes and major diagonal directions at the design wave number k_{opt}. These α's, i.e., (4.17), give a specific algorithm, the stability limit of which is given by (4.13). 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 (4.12) with parameters given in (4.17) along the grid axis direction. That is, the effective time step is defined as

$$\Delta_t' = \frac{\tilde{\omega}}{ck_{\text{opt}}}\Delta_t = \frac{2}{ck_{\text{opt}}}\sin^{-1}\left(S\sin\left(\frac{k_{\text{opt}}\Delta_g}{2}\right)\right). \tag{4.19}$$

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 Figure 4.5. 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 Figure 4.4.

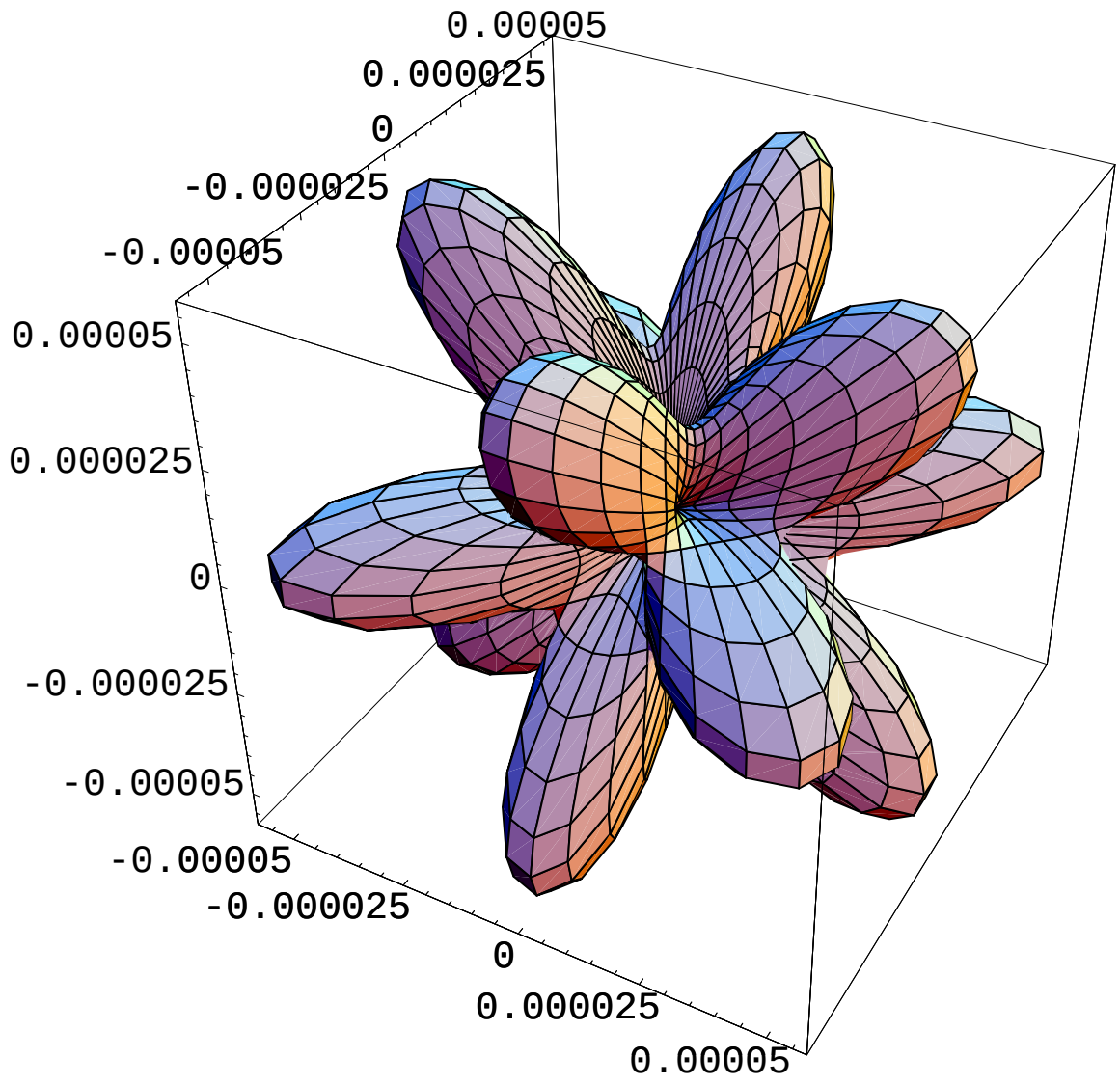

Figure 4.5: Dispersion error surface for the optimized algorithm at 10 samples per wavelength. $\alpha_0' = 0.6651999117$, $\alpha_1' = 0.3348000883$, and $S = 0.866979664$.

4.5 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 (4.8) 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 (4.8) 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 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 (4.8). 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_g in all directions, an analytic expression

can be obtained for k:

$$k(\omega') = \frac{2\sqrt{3}}{\Delta_g} \sin^{-1}\left(\frac{1}{2}\sqrt{3 - \sqrt{9 - \frac{8}{S^2}\sin^2\left(\frac{\omega'\Delta_t}{2}\right)}}\right). \tag{4.20}$$

Multiplying (4.20) by the sound speed c gives the corrected frequency for which the phenomena observed at ω' should be assumed to have occured. Such a correction is demonstrated in the following section.

Depending on the direction and the frequency, the wavenumber obtained from (4.20) may be complex [5]. 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 discretization. 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 frequence for ω'.

4.6 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 (4.12) 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

$$\omega^2 = c^2 \left[\left(\frac{m\pi}{L} \right)^2 + \left(\frac{n\pi}{L} \right)^2 + \left(\frac{p\pi}{L} \right)^2 \right], \tag{4.21}$$

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 (4.21) can be written

$$\omega^2 = c^2 \left(k_x^2 + k_y^2 + k_z^2 \right). \tag{4.22}$$

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

$$k_x = \frac{m\pi}{L}, \qquad k_y = \frac{n\pi}{L}, \qquad k_z = \frac{p\pi}{L}. \tag{4.23}$$

The numerical frequencies that will exist in the FDTD simulation can be found [17] from

(4.12), with the k's from (4.23)

$$\tilde{\omega} = \frac{2}{\Delta_t}\sin^{-1}\Bigg(S\Bigg[\sin^2\left(\frac{m\pi\Delta_g}{2L}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\left(\cos\left(\frac{n\pi\Delta_g}{L}\right) + \cos\left(\frac{p\pi\Delta_g}{L}\right)\right)\right) + \sin^2\left(\frac{n\pi\Delta_g}{2L}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\left(\cos\left(\frac{m\pi\Delta_g}{L}\right) + \cos\left(\frac{p\pi\Delta_g}{L}\right)\right)\right) + \sin^2\left(\frac{p\pi\Delta_g}{2L}\right)\left(\alpha_0 + \frac{\alpha_1}{2}\left(\cos\left(\frac{m\pi\Delta_g}{L}\right) + \cos\left(\frac{n\pi\Delta_g}{L}\right)\right)\right)\Bigg]^{1/2}\Bigg). \tag{4.24}$$

Equation (4.24) 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 $1/2\Delta_t$ [11]. 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 $\Delta_g = 1$ 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 9^3 cells. Because the walls of the resonator are constructed from pressure nodes which are set to zero, the interior length L is $8\Delta_g$ (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 4.6 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 corre-

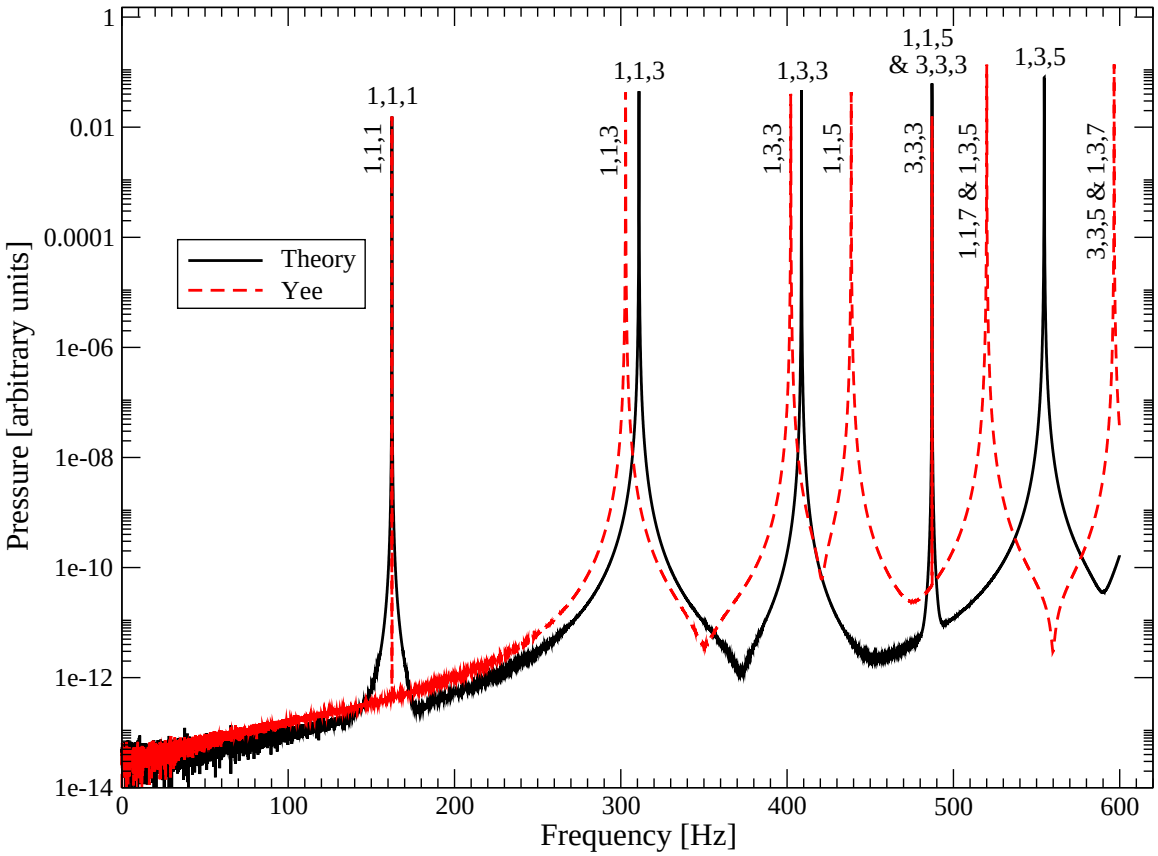

Figure 4.6: Cubic resonator spectrum for Yee algorithm ($\alpha_1 = 0$) where the Courant number is $1/\sqrt{3}$ and the resonator size is $L = 8\Delta_g$. The triplets indicate the mode or modes which correspond to a peak. The horizontal triplets are for the theory (ideal) peaks and the vertical ones 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.

sponds 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 reference [17].

Figure 4.7 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 there are written vertically for the FDTD results. Because of the improved isotropy of the algorithm, the effects of mode splitting 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 4.8 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 Figure 4.8, 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.

4.7 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

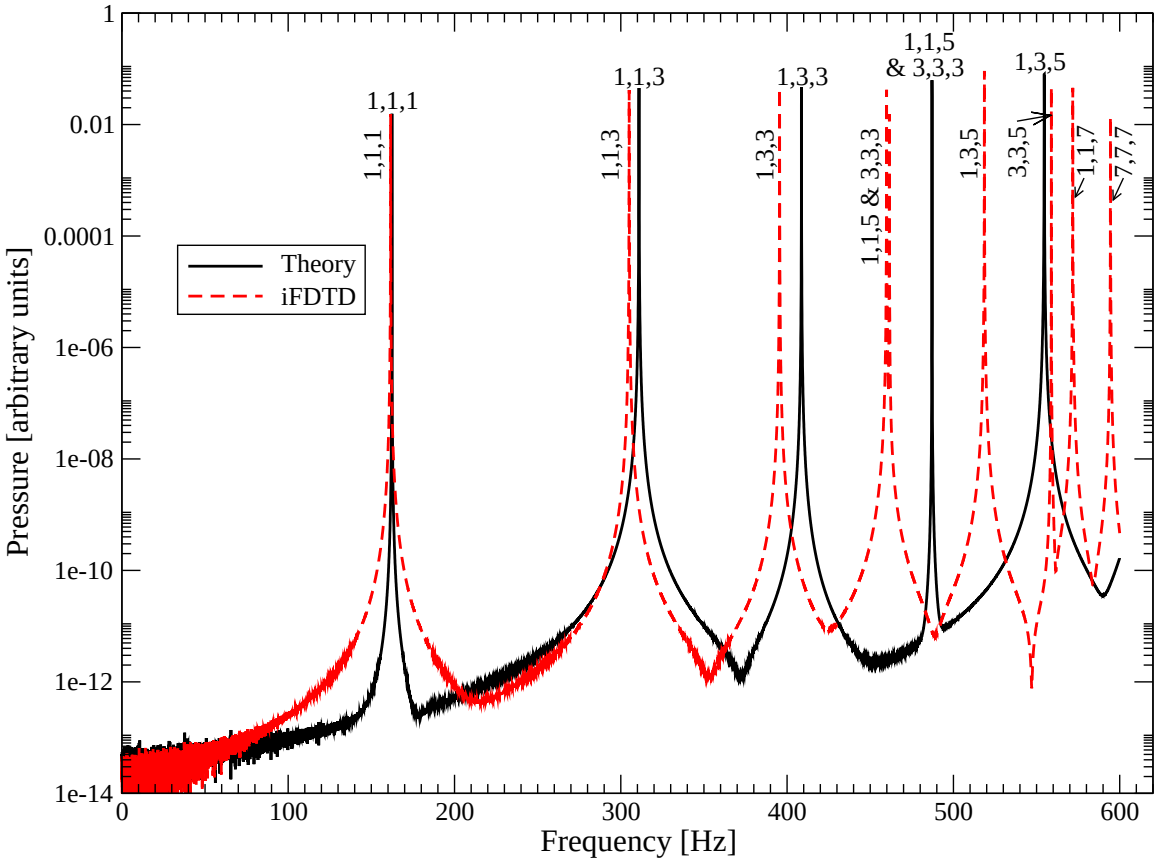

Figure 4.7: Cubic resonator spectrum for proposed isotropic FDTD (iFDTD) algorithm ($\alpha_1 = 1/3$) where the Courant number is $\sqrt{3}/2$ and the resonator size is $L = 8\Delta_g$. The horizontal triplets are for the theory (ideal) peaks and the vertical ones for the iFDTD algorithm. The $(1, 1, 5)$ and $(3, 3, 3)$ modes are 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).

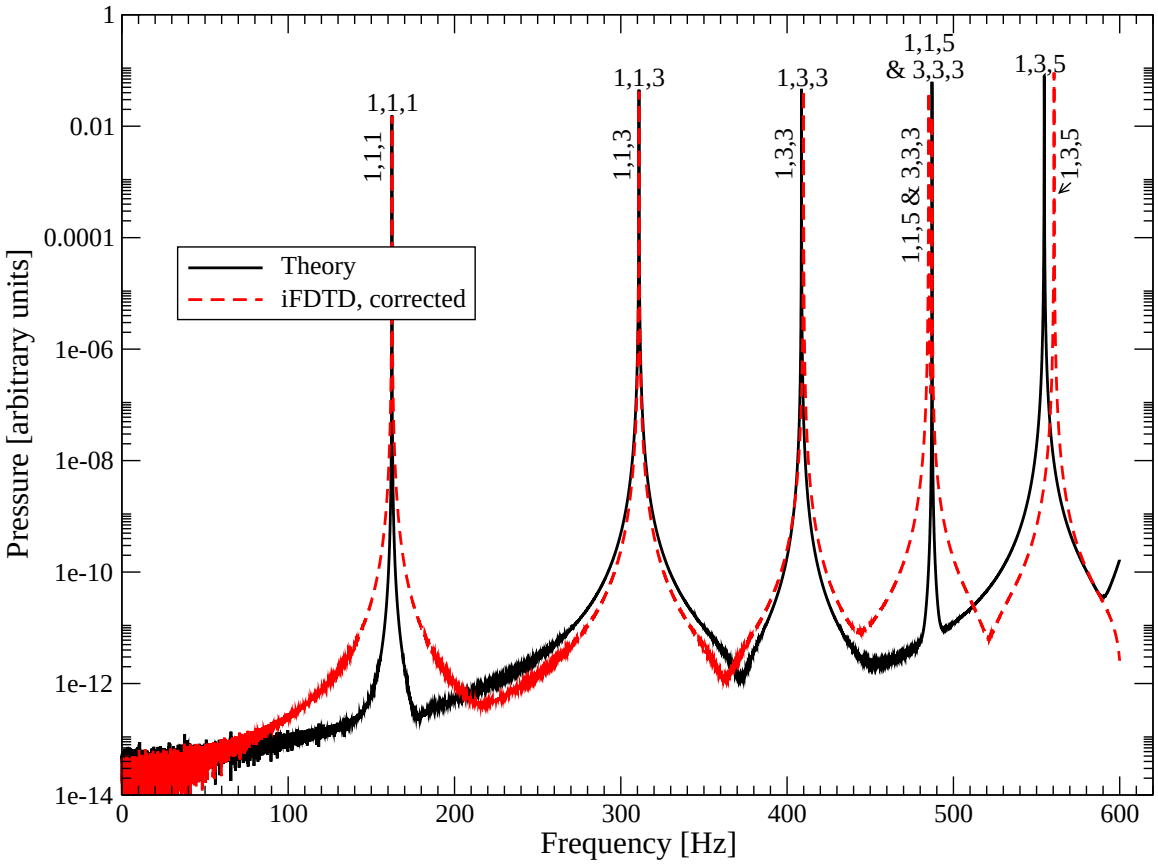

Figure 4.8: Cubic resonator spectrum from Figure 4.7 after applying the post-processing correction described in Section 4.5. The horizontal triplets are for the theory (ideal) peaks and the vertical ones for the iFDTD algorithm. The $(1, 1, 5)$ and $(3, 3, 3)$ modes are still distinct in the iFDTD results since frequency correction cannot recombine modes (but it does serve to narrow the separation between modes which are spuriously split).

is employed. The test geometry is similar to that used in reference [18]. The computational domain is 39^3 cells and the sphere is modeled using eight cells along its radius. Figure 4.9 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, 20, 21]. The grid is terminated with a perfectly-matched layer [22, 23, 24] (PML) which is not shown in Figure 4.9. The PML is eight cells thick and was formulated following the coordinate-stretching approach described by Chew and Weedon [25]. Spectral information is extracted using a discrete Fourier transform [7]. Results are compared to the exact solution over the near-field "evaluation line" which parallels the z axis and is shown in Figure 4.9. Assuming unit-amplitude harmonic ensonification, the exact solution for the scattered pressure is given by [26]

$$P^s = -\sum_{m=0}^{\infty} (-i)^m \, \mathrm{P}_m(\cos\theta) \frac{2m+1}{1+iC_m} \left[\mathrm{j}_m(kR) + i \, \mathrm{n}_m(kR) \right] e^{-i\omega t}, \tag{4.25}$$

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

$$C_m = \frac{\mathrm{j}'_m(k_1 a) \, \mathrm{n}_m(ka) - gh \, \mathrm{j}_m(k_1 a) \, \mathrm{n}'_m(ka)}{\mathrm{j}'_m(k_1 a) \, \mathrm{j}_m(ka) - gh \, \mathrm{j}_m(k_1 a) \, \mathrm{j}'_m(ka)}, \tag{4.26}$$

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 indicated differentiation with respect to the argument so that $\mathrm{j}'_m(kR) = \partial\,\mathrm{j}_m(kR)/\partial(kR)$. The exact solution was calculated using *Mathematica* [27].

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., $c_1 = 2c_0$ 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 4.10 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.

Figure 4.11 shows the scattered field when the sphere has a sound speed half that of the

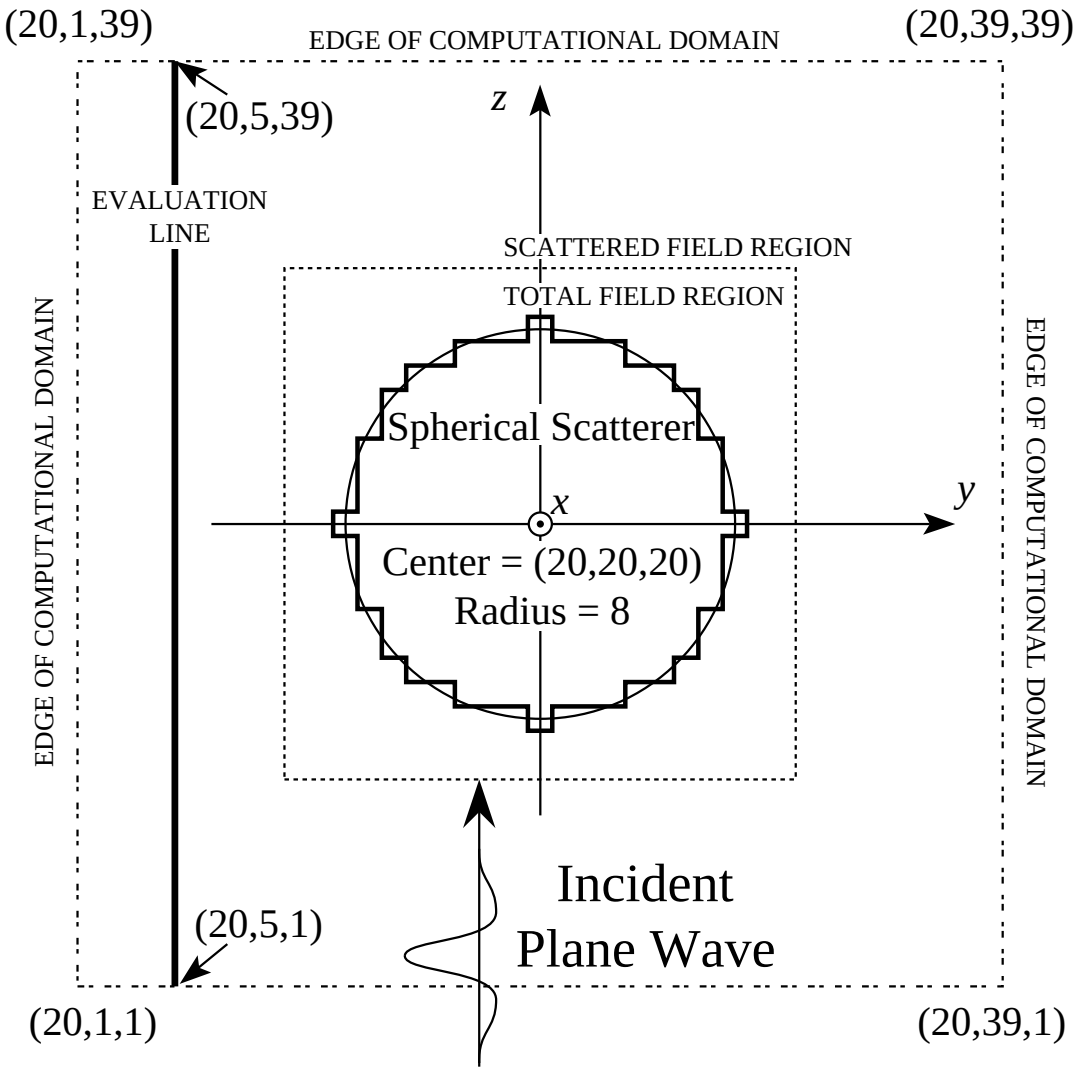

Figure 4.9: Two-dimensional cross-section of the spherical scatterer and the surrounding space. The eight-cell PML used to absorb out-going waves is not shown. All units are in number of cells (thus this cross section was taken over the $x = 20$ plane).

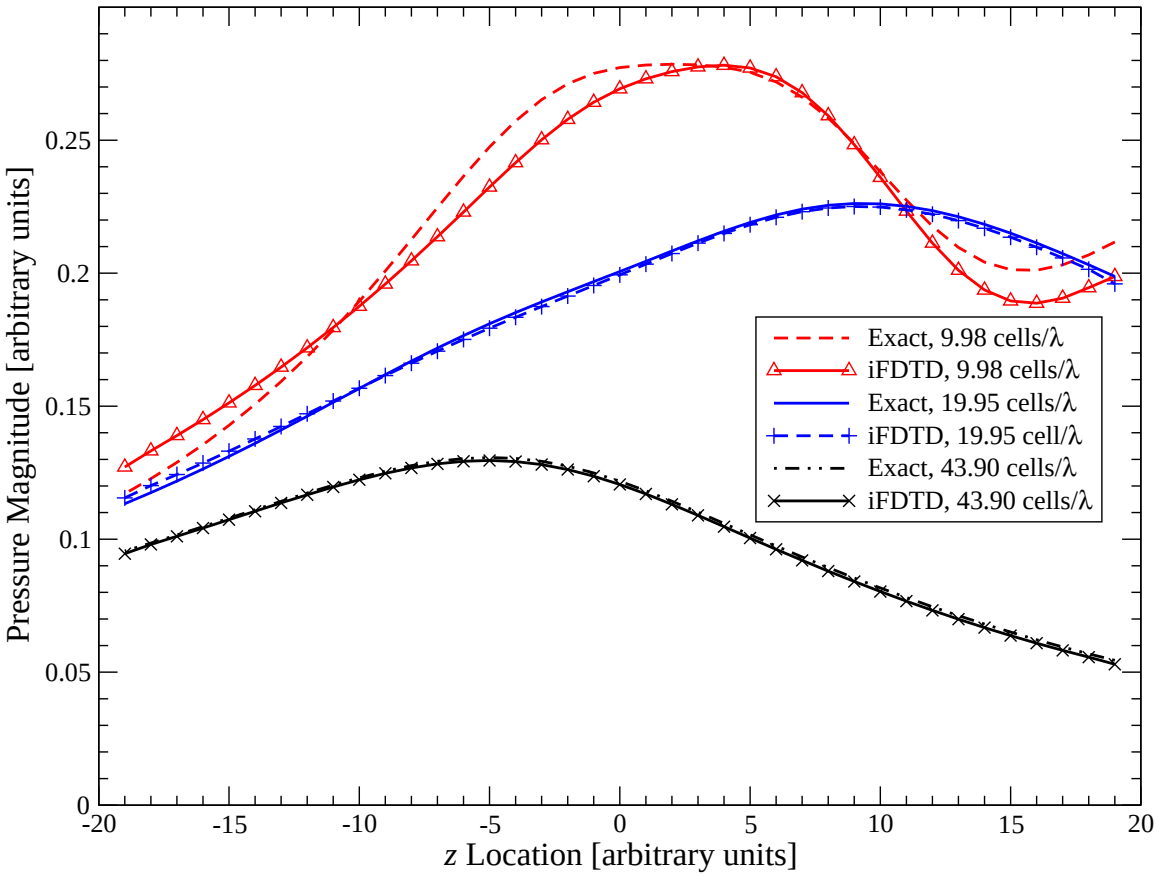

Figure 4.10: Magnitude of pressure versus position at three different frequencies. The sound speed and density of the scatterer are both twice that of the background media, i.e., $c_1 = 2c_0$ and $\rho_1 = 2\rho_0$. The cells per wavelength reported in the legend are as seen in the background media. The isotropic FDTD algorithm proposed in this chapter is indicated by iFDTD.

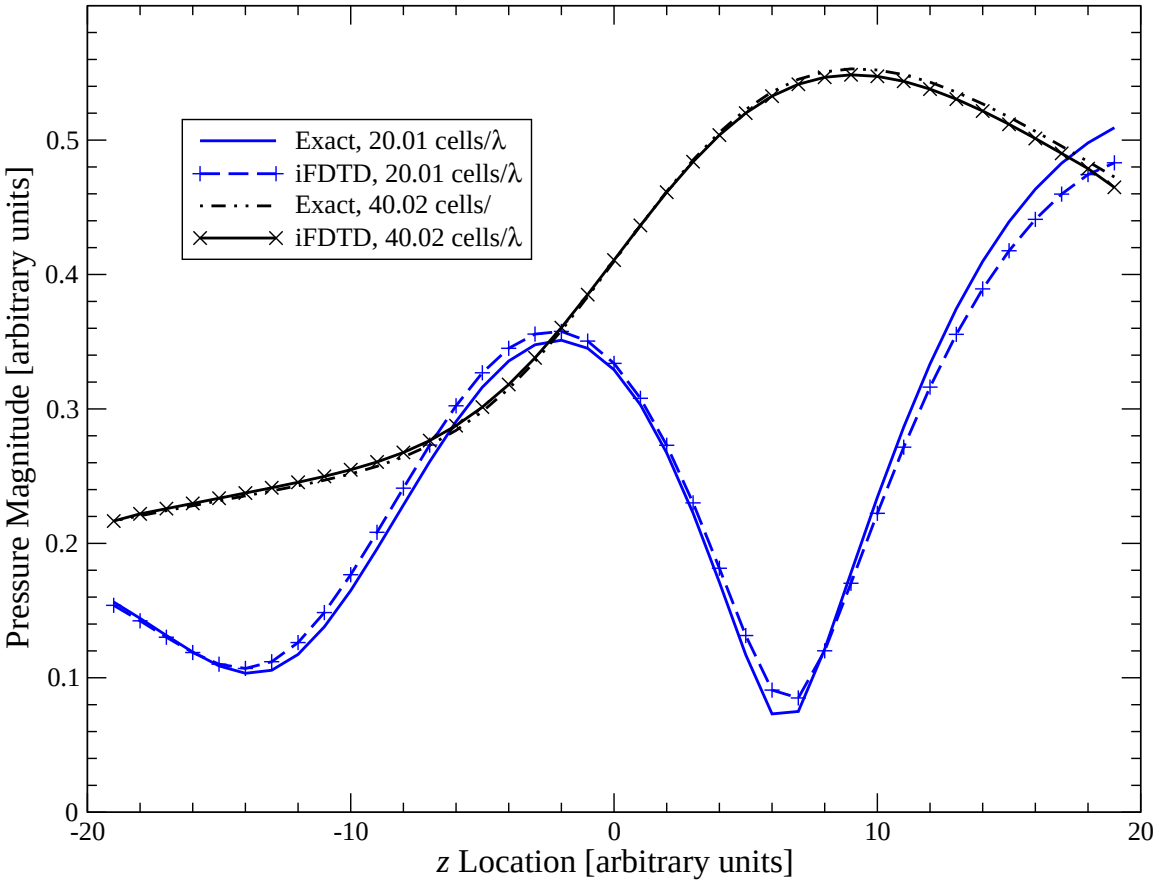

Figure 4.11: Magnitude of pressure versus position at two different frequencies. The sound speed of the scatterer is half that of the background media while the density is constant everywhere, i.e., $c_1 = c_0/2$ and $\rho_1 = \rho_0$. The cells per wavelength reported in the legend are as seen in the background media (the discretization is halved within the scatterer).

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.)

4.8 Chapter Summary

The FDTD algorithm presented in this chapter 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.

Chapter 5

Exact Algorithms

5.1 Introduction

The “magic time step” 1D Yee algorithm is a numerically exact differential equation solver [1, 7]. To obtain theoretically similar performance in 3D we express the divergence, gradient, and curl operators as derivatives with respect to a single variable. The spatial derivative operators presented here provide (theoretically) numerically exact solutions when used with a standard Yee “leap-frog” central difference time derivative. Once the differential operators are defined a dispersion relation is derived for hyperbolic systems of two coupled first-order equations and specific dispersion relations for 3D acoustic and electromagnetics algorithms are then given. With the specified choice of algorithm parameters, the dispersion analysis shows theoretically exact propagation. An exact theory can be used as the basis for designing practical approximations. As a test of these volumetric differential operators, acoustic and electromagnetic algorithms have been coded. The performance of these algorithms and the classic Yee algorithm are shown for the resonator test cases.

5.2 Spherical Volume Differential Operators

The spherical differential operators needed for exact algorithms can be defined as follows. Start with the gradient identity, and the closely-related Gauss's divergence and vector Stokes' theorems:

$$\iiint_{\mathrm{B}} \boldsymbol{\nabla}\psi(\mathbf{r}+\mathbf{r}')\,\mathrm{d}v = \iint_{\partial\mathrm{B}} \hat{\mathbf{n}}\psi(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s, \tag{5.1}$$

$$\iiint_{\mathrm{B}} \boldsymbol{\nabla}\cdot\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}v = \iint_{\partial\mathrm{B}} \hat{\mathbf{n}}\cdot\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s, \tag{5.2}$$

$$\iiint_{\mathrm{B}} \boldsymbol{\nabla}\times\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}v = \iint_{\partial\mathrm{B}} \hat{\mathbf{n}}\times\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s, \tag{5.3}$$

where primes indicate variables of integration when necessary, and $\mathbf{r}$ is the position about which the integrals are computed. Assume the fields, surfaces, and volumes satisfy the conditions necessary for the theorems (5.1)–(5.3) to hold. Choose a sphere of radius $\rho \geq 0$ as the shape of the volume B. Now define the spherical gradient, divergence and curl as the scalar derivative of the corresponding integral identity

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r}) = \frac{\partial}{\partial V}\iiint_{\mathrm{B}} \boldsymbol{\nabla}\psi(\mathbf{r}+\mathbf{r}')\,\mathrm{d}v = \frac{\partial}{\partial V}\iint_{\partial\mathrm{B}} \hat{\mathbf{n}}\psi(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s, \tag{5.4}$$

$$\boldsymbol{\nabla}_{\mathrm{s}}\cdot\mathbf{W}(\mathbf{r}) = \frac{\partial}{\partial V}\iiint_{\mathrm{B}} \boldsymbol{\nabla}\cdot\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}v = \frac{\partial}{\partial V}\iint_{\partial\mathrm{B}} \hat{\mathbf{n}}\cdot\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s, \tag{5.5}$$

$$\boldsymbol{\nabla}_{\mathrm{s}}\times\mathbf{W}(\mathbf{r}) = \frac{\partial}{\partial V}\iiint_{\mathrm{B}} \boldsymbol{\nabla}\times\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}v = \frac{\partial}{\partial V}\iint_{\partial\mathrm{B}} \hat{\mathbf{n}}\times\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s, \tag{5.6}$$

where the derivative is with respect to the volume V of sphere B, $V = \frac{4}{3}\pi\rho^3$. The symbol $\boldsymbol{\nabla}_{\mathrm{s}}$ is used to denote a nabla operator that measures over a finite spherical volume (not the

usual point operator). For analysis either the volume or surface integrals in (5.4)–(5.6) can be used as required, but in algorithms only the surface integrals are used. With the choice of spherical volumes, we can for illustrative purposes write out details of the right hand side of (e.g.) (5.4)

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r}) = \frac{1}{4\pi\rho^2}\frac{\partial}{\partial\rho}\iint_{\partial\mathrm{B}}(\sin\theta\cos\phi\hat{\mathbf{x}} + \sin\theta\sin\phi\hat{\mathbf{y}} + \cos\theta\hat{\mathbf{z}})$$
$$\psi((x+\rho\sin\theta\cos\phi)\hat{\mathbf{x}} + (y+\rho\sin\theta\sin\phi)\hat{\mathbf{y}} + (z+\rho\cos\theta)\hat{\mathbf{z}})\rho^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi, \tag{5.7}$$

where

$$\begin{aligned}\mathbf{r} &= r\hat{\mathbf{r}} = x\hat{\mathbf{x}} + y\hat{\mathbf{y}} + z\hat{\mathbf{z}},\\ \mathbf{r}' &= \rho\hat{\mathbf{n}} = \rho(\sin\theta\cos\phi\hat{\mathbf{x}} + \sin\theta\sin\phi\hat{\mathbf{y}} + \cos\theta\hat{\mathbf{z}}),\end{aligned} \tag{5.8}$$

and where the variables of integration are the spherical polar angle θ and the equatorial angle ϕ.

5.2.1 Correspondence of Spherical and Point Derivatives

To see that the spherical derivatives correspond to the usual point derivatives when the sphere size is zero, write the volume partial derivative as a limit and let $F(V,\mathbf{r}) = \iint \hat{\mathbf{n}}\cdot\mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s$ in (5.5)

$$\boldsymbol{\nabla}_{\mathrm{s}}\cdot\mathbf{W}(\mathbf{r}) = \lim_{v\to 0}\frac{F(V+v,\mathbf{r}) - F(V,\mathbf{r})}{v} \qquad V, v \geq 0, \tag{5.9}$$

then set the volume V to zero

$$\begin{aligned}\boldsymbol{\nabla}_{\mathrm{s}} \cdot \mathbf{W}(\mathbf{r})\big|_{V=0} &= \lim_{v\to 0} \frac{F(v,\mathbf{r}) - F(0,\mathbf{r})}{v} = \lim_{v\to 0} \frac{F(v,\mathbf{r})}{v} \\ &= \lim_{v\to 0} \frac{1}{v} \iint_{\partial \mathrm{B}} \hat{\mathbf{n}} \cdot \mathbf{W}(\mathbf{r}+\mathbf{r}')\,\mathrm{d}s = \boldsymbol{\nabla} \cdot \mathbf{W}(\mathbf{r}).\end{aligned} \tag{5.10}$$

The final line of (5.10) is a typical definition of the usual point divergence [28], "typical" because the usual point divergence definition does not need to specify the shape. The volume operator $\boldsymbol{\nabla}_{\mathrm{s}}$ is useful because it is the usual nabla operator when computed with zero radius, while for non-zero radius the operator can produce exact time-domain algorithms. Equations (5.4)–(5.6) are the definitions of the space derivative operators for the present class of exact algorithms. As written, (5.4)–(5.6) are functions of the size of the sphere. For some particular algorithm, the as yet undetermined radius of the sphere is fixed.

5.3 Stability and Dispersion Analysis

The dispersion relation is obtained in the same manner as in Section 2.2. Here however, the spatial derivatives are given by (5.4)–(5.6).

5.3.1 Effect on Plane Waves

In order to derive the algorithm stability properties and dispersion relation the effect of the spherical volume differential operators on plane waves is required. For simplicity a cubic cell grid of step size Δ_g will be used throughout. For a specific implementation of an algorithm a spatial grid will need to be chosen (e.g., staggered or collocated), at this point we only require a staggered-in-time grid. The staggering or collocation of the spatial grid

need not be specified yet for the volume operators. Given a complex scalar plane wave with wave vector $\mathbf{k}$ and frequency ω

$$\psi(\mathbf{r}, t) = e^{i(\mathbf{k}\cdot\mathbf{r} - \omega t)}, \tag{5.11}$$

where

$$\mathbf{k} = k\hat{\mathbf{k}} = k(\sin\alpha\cos\beta\hat{\mathbf{x}} + \sin\alpha\sin\beta\hat{\mathbf{y}} + \cos\alpha\hat{\mathbf{z}}) = k_x\hat{\mathbf{x}} + k_y\hat{\mathbf{y}} + k_z\hat{\mathbf{z}}, \tag{5.12}$$

the volume derivatives and the usual Yee time derivative are computed. The angles α and β are the spherical polar and equatorial angles of the wave vector. The classic Yee algorithm uses central differences to approximate the temporal derivatives. Thus in the Yee algorithm the temporal derivative of a plane wave is given by (2.3), repeated here for convenience is

$$D_t\psi(\mathbf{r}, t) = -\frac{2i}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right)\psi(\mathbf{r}, t). \tag{5.13}$$

The space derivative operators for an exact algorithm must have the same functional effect on the 3D plane wave as (5.13) in order to obtain a functionally 1D dispersion relation for the 3D space. Then it will be possible to choose values of the algorithm parameters to obtain matched space and time derivatives, giving theoretically exact homogeneous-space propagation.

When acting on a plane wave (5.11), the spherical volume gradient (5.4) becomes

$$\begin{aligned}\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r},t) &= \frac{1}{4\pi\rho^2}\frac{\partial}{\partial\rho}\left(\int_0^{2\pi}\int_0^{\pi}\hat{\mathbf{n}}\psi(\mathbf{r}+\mathbf{r}',t)\rho^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi\right)\\ &= \frac{1}{4\pi\rho^2}\frac{\partial}{\partial\rho}\left(\rho^2\int_0^{2\pi}\int_0^{\pi}\hat{\mathbf{n}}e^{i\mathbf{k}\cdot\mathbf{r}'}\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi\right)\psi(\mathbf{r},t).\end{aligned} \tag{5.14}$$

The integrals in (5.14) express the standard relation between spherical Bessel functions and surface integrals of spherical harmonics. See for example Stratton [29, Sec. 7.7, Eq. (60)]. The spherical Bessel function integral is

$$i^n 4\pi\,\mathrm{j}_n(k\rho)\,\mathrm{P}_n^m(\cos\alpha)\begin{matrix}\sin(m\beta)\\ \cos(m\beta)\end{matrix} = \int_0^{2\pi}\int_0^{\pi} e^{i\mathbf{k}\cdot\mathbf{r}'}\,\mathrm{P}_n^m(\cos\theta)\begin{matrix}\sin(m\phi)\\ \cos(m\phi)\end{matrix}\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi, \tag{5.15}$$

where j_n is the spherical Bessel function of order n, and P_n^m is the associated Legendre polynomial. Using (5.15) and (5.12), the integrals in (5.14) become

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r},t) = \frac{1}{4\pi\rho^2}\frac{\partial}{\partial\rho}\left(i4\pi\rho^2\,\mathrm{j}_1(k\rho)\right)\hat{\mathbf{k}}\psi(\mathbf{r},t). \tag{5.16}$$

The derivative is best handled by using the spherical Bessel differentiation formula [30, Eq. (10.1.23)]

$$\left(\frac{d}{zdz}\right)^m\left[z^{n+1}\,\mathrm{j}_n(z)\right] = z^{n-m+1}\,\mathrm{j}_{n-m}(z). \tag{5.17}$$

By application of (5.17) to (5.16), one obtains the effect on the plane wave of the volume gradient operator

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r},t) = i\,\mathrm{j}_0(k\rho)\,\mathbf{k}\,\psi(\mathbf{r},t). \tag{5.18}$$

Now set the radius $\rho = a\Delta_g$, where a is the radius of the operator sphere in grid steps. With all the terms of the volume gradient found, the various integrals in the volume divergence and volume curl have been solved for vector plane waves. The volume gradient, divergence, and curl of some scalar ψ or vector $\mathbf{A}$ plane wave become

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r},t) = i\,\mathrm{j}_0(ka\Delta_g)\,\mathbf{k}\,\psi(\mathbf{r},t), \tag{5.19}$$

$$\boldsymbol{\nabla}_{\mathrm{s}}\cdot\mathbf{A}(\mathbf{r},t) = i\,\mathrm{j}_0(ka\Delta_g)\,\mathbf{k}\cdot\mathbf{A}(\mathbf{r},t), \tag{5.20}$$

$$\boldsymbol{\nabla}_{\mathrm{s}}\times\mathbf{A}(\mathbf{r},t) = i\,\mathrm{j}_0(ka\Delta_g)\,\mathbf{k}\times\mathbf{A}(\mathbf{r},t). \tag{5.21}$$

The key feature of the results (5.19)–(5.21) is the $\mathrm{j}_0(ka\Delta_g)$ factor. As will be shown, this factor allows time-domain algorithms constructed with these volume operators to be exact. Note that if the algorithm radius a is zero, then (5.19)–(5.21) are the usual gradient, divergence, and curl of a plane wave, as expected from the previously established correspondence of zero-sized spherical derivatives and point derivatives.

5.3.2 Dispersion Relation for Exact Algorithms

The operators presented here can be used to construct numerically exact algorithms for hyperbolic systems of coupled first-order equations, for example Maxwell's equations. The time derivatives will be approximated with the usual "leap-frog" central difference as used in the classical Yee FDTD algorithm [1], with the volume divergence, volume gradient, or volume curl given by (5.4)–(5.6). Given a self-consistent hyperbolic system of two coupled

first-order equations

$$\frac{\partial F_1}{\partial t} = h_1 \triangle_1 F_2, \quad \frac{\partial F_2}{\partial t} = h_2 \triangle_2 F_1, \tag{5.22}$$

where the $\triangle_i$'s are one of div, grad, or curl and the h_i's are the given scale constants for each equation. The system must have propagating solutions. A plane wave solution $e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t)}$ is assumed (vector or scalar, as required), and the numerical differentiation on the plane wave as given by (5.19)–(5.21). The usual finite-difference time derivatives (5.13) are substituted into (5.22) and the resulting system is reduced and simplified. There are three possible forms for each of the system's equations

$$\frac{\partial F_i}{\partial t} = h_i \boldsymbol{\nabla} \cdot \mathbf{F}_j, \qquad \frac{\partial \mathbf{F}_m}{\partial t} = h_m \boldsymbol{\nabla} F_n, \qquad \frac{\partial \mathbf{F}_p}{\partial t} = h_p \boldsymbol{\nabla} \times \mathbf{F}_q, \tag{5.23}$$

where the F's, represent the fields of the system. The finite-difference time derivative (5.13) and the spherical volume derivatives (5.19)–(5.21) acting on (5.23) with assumed plane wave solutions will produce:

$$\begin{aligned} -\frac{2}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right) F_i &= h_i \, \mathrm{j}_0(ka\Delta_g)\mathbf{k}\cdot\mathbf{F}_j, \\ -\frac{2}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right) \mathbf{F}_m &= h_m \, \mathrm{j}_0(ka\Delta_g)\mathbf{k} F_n, \\ -\frac{2}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right) \mathbf{F}_p &= h_p \, \mathrm{j}_0(ka\Delta_g)\mathbf{k}\times\mathbf{F}_q. \end{aligned} \tag{5.24}$$

Since the system is hyperbolic with propagating solutions, the dispersion relation results from simultaneously solving the algebraic system of equations produced by substituting

(5.24) into (5.22) and eliminating common factors. This yields

$$\left(\frac{2}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right)\right)^2 = (h_1 h_2\, k\, \mathrm{j}_0(ka\Delta_g))^2. \tag{5.25}$$

Substituting the trigonometric form for the spherical Bessel function into (5.25) and simplifying yields

$$\left(\frac{2}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right)\right)^2 = \left(h_1 h_2 \frac{\sin(ka\Delta_g)}{a\Delta_g}\right)^2. \tag{5.26}$$

Note that, by design, this is functionally equivalent to the Yee 1D dispersion relation [7]. The algorithm parameters (Δ_t, Δ_g, a) are chosen such that

$$\left(\frac{h_1 h_2 \Delta_t}{2a\Delta_g}\right)^2 = 1, \tag{5.27}$$

or equivalently, so that the Yee algorithm's Courant number, S, is

$$S = \frac{c\Delta_t}{\Delta_g} = \frac{|h_1 h_2|\Delta_t}{\Delta_g} = 2a, \tag{5.28}$$

where $c = |h_1 h_2|$ is the wave speed. With this choice of parameters the coefficients of the sine functions in (5.26) are unity. Taking the square-root and the arcsine of both sides of (5.26) and simplifying produces

$$\omega = \sqrt{|h_1 h_2|}\, k = ck, \tag{5.29}$$

which is the ideal homogeneous space continuum result. In order for (5.29) to be true we must also have spatially and temporally band-limited waves because the computation is sampled in space and time:

$$\omega \leq \frac{\pi}{\Delta_t} \qquad \text{and} \qquad |k_x|, |k_y|, |k_z| \leq \frac{\pi}{\Delta_g}. \tag{5.30}$$

A surprising aspect of the stability and exactness condition (5.27) is that it can be imposed for any Δ_t because the radius a is a free parameter of the algorithm. It is unknown if practical versions of this algorithm work for large time steps. Other algorithms which are stable for arbitrary time steps are known, for example the "One-Step" method of De Raedt et. al. [31].

5.3.3 Acoustics Dispersion Relation

The first specific example of a dispersion relation is for acoustics. For homogeneous-space small-signal acoustics the governing equations are

$$\frac{\partial \mathbf{V}}{\partial t} = -\frac{1}{\varrho}\boldsymbol{\nabla} P, \qquad \frac{\partial P}{\partial t} = -\varrho c^2 \boldsymbol{\nabla} \cdot \mathbf{V}, \tag{5.31}$$

where ϱ is the density, c is the wave speed, and $\mathbf{V}$ and P are the velocity and pressure fields, respectively. Applying (5.13), (5.19), and (5.20) to each term of (5.31) for assumed

plane wave solutions with wave vector $\mathbf{k}$ and frequency ω yields

$$\begin{aligned} -\frac{2i}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right)\mathbf{V} &= -\frac{1}{\varrho}i\,\mathrm{j}_0(ka\Delta_g)\mathbf{k}P, \\ -\frac{2i}{\Delta_t}\sin\left(\frac{\omega\Delta_t}{2}\right)P &= -\varrho c^2 i\,\mathrm{j}_0(ka\Delta_g)\mathbf{k}\cdot\mathbf{V}. \end{aligned} \tag{5.32}$$

Solving (5.32) simultaneously to eliminate (for example) $\mathbf{V}$ yields

$$\frac{4}{\Delta_t^2}\sin^2\left(\frac{\omega\Delta_t}{2}\right)P = c^2\,\mathrm{j}_0^2(ka\Delta_g)\mathbf{k}\cdot\mathbf{k}P. \tag{5.33}$$

Simplifying and eliminating the remaining common factors produces the dispersion relation

$$\frac{4}{\Delta_t^2}\sin^2\left(\frac{\omega\Delta_t}{2}\right) = c^2k^2\,\mathrm{j}_0^2(ka\Delta_g) = \frac{c^2}{(a\Delta_g)^2}\sin^2(ka\Delta_g). \tag{5.34}$$

This is the same as the Yee 1D dispersion equation, so the previous stability and exactness conditions (5.27) and sampling limits (5.30) apply. Then using these conditions the dispersion relation becomes

$$\omega = ck, \tag{5.35}$$

which is the ideal result up to the sampling limits.

5.3.4 Electromagnetics Dispersion Relation

We obtain the electromagnetics dispersion relation with similar ease. Starting with Maxwell's curl equations in homogeneous source-free space

$$\epsilon \frac{\partial \mathbf{E}}{\partial t} = \boldsymbol{\nabla} \times \mathbf{H}, \qquad \mu \frac{\partial \mathbf{H}}{\partial t} = -\boldsymbol{\nabla} \times \mathbf{E}, \tag{5.36}$$

where $\mathbf{E}, \mathbf{H}$ are the electric and magnetic fields, and ϵ, μ are the permittivity and permeability. Applying (5.13) and (5.21) to each term of (5.36) for an assumed vector plane wave solution yields

$$\begin{aligned} -\frac{2i\epsilon}{\Delta_t} \sin\left(\frac{\omega\Delta_t}{2}\right)\mathbf{E} &= i\, \mathrm{j}_0(ka\Delta_g)\mathbf{k} \times \mathbf{H}, \\ -\frac{2i\mu}{\Delta_t} \sin\left(\frac{\omega\Delta_t}{2}\right)\mathbf{H} &= -i\, \mathrm{j}_0(ka\Delta_g)\mathbf{k} \times \mathbf{E}. \end{aligned} \tag{5.37}$$

Eliminating one of the fields, chosen here to be $\mathbf{H}$, produces a wave equation in the remaining field

$$\frac{4}{c^2\Delta_t^2} \sin^2\left(\frac{\omega\Delta_t}{2}\right)\mathbf{E} = -\,\mathrm{j}_0^2(ka\Delta_g)\,\mathbf{k} \times (\mathbf{k} \times \mathbf{E}) = \mathrm{j}_0^2(ka\Delta_g)\,k^2\mathbf{E}, \tag{5.38}$$

where $c = 1/\sqrt{\epsilon\mu}$ is the wave speed. Eliminating the remaining common factors gives the dispersion relation

$$\frac{4}{c^2\Delta_t^2} \sin^2\left(\frac{\omega\Delta_t}{2}\right) = k^2\, \mathrm{j}_0^2(ka\Delta_g) = \frac{1}{(a\Delta_g)^2} \sin^2(ka\Delta_g). \tag{5.39}$$

Once again this is formally the one-dimensional Yee dispersion relation, so the stability and exactness conditions (5.27) and sampling limits (5.30) apply, permitting theoretically ideal propagation up to the sampling limits.

5.4 Proof-of-Principle Algorithms

In order to test the concepts of the volume operators presented here, a canonical problem is solved analytically and numerically with the classical Yee and volume algorithms. The test case is a cubic resonator. In the scalar-field (acoustics) case the boundaries are pressure-release (i.e., Dirichlet boundary conditions on the pressure), for the vector-field (electromagnetics) case the boundary is a perfect electric conductor (PEC). Cubic resonators are canonical structures which provide instructive algorithm performance test beds. Furthermore the domains are terminated with boundary conditions suitable for the problem, thus no absorbing boundaries are required. The test resonators are excited and sampled at the center of the cubic domain, the time series is Fourier transformed with a raised-cosine window ($1 - \cos(2\pi n \Delta_t / T)$, where T is the total simulation time) to obtain amplitude spectra. The material parameters, wave speed, and space delta are all set to unity. All calculations are done with single precision (32 bit) floating point numbers. The computations are run for 65536 time steps—which is sufficient to see the approximately 10^7 dynamic range of the 32 bit computations. The spectra are plotted from zero frequency to the Nyquist limit. To facilitate meaningful comparisons, all algorithms are implemented on the usual Yee staggered cubic-cell grid with $c\Delta_t/\Delta_g = 1/\sqrt{3}$.

5.4.1 Sampled Resonator Description

The resonators used here are cubic, with a volume of L^3 cubic meters. This is discretized with $M = 10$ cells in each axis, so $L = M\Delta_g$. The source and sample point is centered in the resonator. The resonator walls are aligned with the grid.

For acoustics the source and sample point is the pressure node at the center of the resonator. The acoustics grid will have $(M+1)^3$ pressure (P) nodes, the outer surface of P nodes are initialized to zero, and remain zero. Velocity nodes are staggered between the P nodes in the usual acoustics Yee grid. In order to have a P node at the geometric center of the resonator, M must be even.

The electromagnetic source and sample is the two center-most E_z nodes, two nodes are required because of the symmetries of the electromagnetics Yee grid. The E_z grid will have $(M+1)^2M$ nodes. The E_z nodes are zero on the x constant and y constant boundary surfaces, there are no E_z nodes in the z constant boundary surfaces. Centering the source and sample again requires M even. The other **E** components, and the **H** components are staggered around E_z in the usual electromagnetics Yee grid.

5.4.2 Reference Theory

The reference theories are analytic resonators with band-limited $\mathbf{k}$ and ω (5.30), where the band-limiting accounts for the sampled nature of the simulations. The eigenfunction sums are truncated to accomplish the band-limiting. The band limited solutions are therefore approximate solutions to the given problems. A sampled analytic theory can allow prediction of the behavior of ideal sampled numerical calculations. The coordinate sys-

tems for analysis are chosen so that the origin is at the center of the cubic resonators, with walls aligned with coordinate surfaces. The interior region of the resonators will be denoted as Ω, the boundary by $\partial\Omega$, and the closure by $\bar{\Omega}$.

For both the electromagnetics and acoustics problems the theoretical solution for the time evolution of the field at the sample location is Fourier transformed and plotted for comparison with the simulation spectra.

5.4.3 Acoustics Resonator Analysis

The Yee or volume-derivative algorithms use the coupled first order system to calculate the fields in the resonator

$$\frac{\partial \mathbf{V}}{\partial t} = -\frac{1}{\varrho}\boldsymbol{\nabla} P, \qquad \frac{\partial P}{\partial t} = -\varrho c^2 \boldsymbol{\nabla}\cdot\mathbf{V} + J. \tag{5.40}$$

For analysis the theoretical acoustics resonator is solved as a forcing-function problem, where the wave equation is obtained from system (5.40) assuming material parameters are constants. Here J is an additive pressure point-source at the origin. The problem is then

$$-c^2\nabla^2 P(\mathbf{r},t) + \frac{\partial^2 P(\mathbf{r},t)}{\partial t^2} = \frac{\partial J(t)}{\partial t}\delta(\mathbf{r}), \qquad \mathbf{r}\in\Omega\,,\ t\geq 0, \tag{5.41}$$

$$P(\mathbf{r},t)\big|_{\partial\Omega} = 0, \qquad \mathbf{r}\in\partial\Omega\,,\ t\geq 0, \tag{5.42}$$

$$P(\mathbf{r},t)\equiv 0, \qquad J(t)\equiv 0, \qquad \mathbf{r}\in\bar{\Omega}\,,\ t<0, \tag{5.43}$$

where $\delta(\mathbf{r})$ is a delta function locating the source (and sample) point at the center of the resonator. For analysis, $\delta()$ is a Dirac delta, in the discrete simulations $\delta()$ is a Kronecker or

discrete delta. Partial differential equation (PDE) (5.41) is the governing equation, (5.42) is the boundary condition, and (5.43) is a causality condition. The solution is given by

$$\begin{aligned} P(\mathbf{r},t) &= \sum_{\substack{m,p,q=1 \\ \text{ODD}}}^{\infty} A_{mpq}(t)\Phi_{mpq}(\mathbf{r}) \\ &= \sum_{\substack{m,p,q=1 \\ \text{ODD}}}^{\infty} A_{mpq}(t)\cos\left(\frac{m\pi x}{L}\right)\cos\left(\frac{p\pi y}{L}\right)\cos\left(\frac{q\pi z}{L}\right), \end{aligned} \tag{5.44}$$

where $\Phi_{mpq}(\mathbf{r})$ are the eigenfunctions and $A_{mpq}(t)$ is the mode time dependence. Due to the symmetries and boundary conditions of the acoustics problem, the mode indexes m, p, q are odd.

The forcing function (right hand side of (5.41)) is expanded in the eigenfunctions

$$F(\mathbf{r},t) = \frac{\partial J(t)}{\partial t}\delta(\mathbf{r}) = \sum_{\substack{m,p,q=1 \\ \text{ODD}}}^{\infty} B_{mpq}(t)\Phi_{mpq}(\mathbf{r}). \tag{5.45}$$

Now take the source time dependence to be $J(t) = \delta(t)$, then the $B_{mpq}(t)$ can be found in the usual way with

$$B_{mpq}(t) = \frac{\iiint_\Omega F(\mathbf{r},t)\Phi_{mpq}(\mathbf{r})\,\mathrm{d}v}{\iiint_\Omega \Phi^2_{mpq}(\mathbf{r})\,\mathrm{d}v} = \frac{8}{L^3}\delta'(t). \tag{5.46}$$

Then the $A_{mpq}(t)$ coefficients are found by substituting (5.44) and (5.45) into (5.41), giving an ordinary differential equation (ODE)

$$\omega^2_{mpq}A_{mpq}(t) + A''_{mpq}(t) = B_{mpq}(t), \tag{5.47}$$

the solution, satisfying (5.41)–(5.43) is

$$A_{mpq}(t) = \frac{8}{L^3} H(t) \cos(\omega_{mpq} t), \tag{5.48}$$

where $H(t)$ is a step function. The eigenvalues or resonant mode frequencies are

$$k_{mpq}^2 = \left(\frac{\pi}{L}\right)^2 \left(m^2 + p^2 + q^2\right) = \left(\frac{\omega_{mpq}}{c}\right)^2. \tag{5.49}$$

The solution to problem (5.41)–(5.43) is

$$P(\mathbf{r}, t) = H(t) \frac{8}{L^3} \sum_{\substack{m,p,q=1 \\ \text{ODD}}}^{\infty} \cos(\omega_{mpq} t) \Phi_{mpq}(\mathbf{r}). \tag{5.50}$$

To band limit the solution, the sum is truncated. Because of the sampling considerations, the mode indexes must all satisfy $m, p, q < M$. We have discretized with $M = 10$ cells, so the sums run to $M - 1 = 9$. At the sample point in the center of the domain, the field is then

$$P(\mathbf{0}, t) = H(t) \frac{8}{L^3} \sum_{\substack{m,p,q=1 \\ \text{ODD}}}^{M-1} \cos(\omega_{mpq} t) = \frac{H(t)}{(M/2)^3 \Delta_g^3} \sum_{\substack{m,p,q=1 \\ \text{ODD}}}^{M-1} \cos(\omega_{mpq} t), \tag{5.51}$$

where $(M/2)^3 = 125$ is the total number of modes, including degeneracies.

The source function for the acoustic simulation is an additive pressure source two Δ_t units long with amplitude $1/2$. Although it is not obvious that this is the proper source to simulate the specified theoretical problem (the obvious choice being a unit amplitude pulse

one Δ_t unit long), this source implementation is a consequence of sampling considerations. Two samples per pulse are required. This source issue can be demonstrated as follows. In addition to being able to construct the time evolution of the fields in an analytic acoustics resonator, as was shown in Section 5.4.3, it is also possible to construct the time evolution that will be produced by a Yee simulation without actually having to run the FDTD simulation. As shown in [17], given the complete analytical solution of the sampled resonator, the frequencies that will exist in the numerical grid can be precisely predicted by using the dispersion relation. After dispersion shifting there will be a new list of frequencies and amplitudes. This new list is the set of basis functions for generating a dispersion-shifted theoretical prediction of the algorithm behavior. There is also a source amplitude issue. The theoretical solution (5.51) assumed a unit amplitude delta function as the J in (5.40). In the FDTD method a unit amplitude discrete delta signal acts over the entire unit-cell space-time volume, not at a space-time point. We then need to either divide the signal in the FDTD computation by $\Delta_t \Delta_g^3$, or multiply the prediction (5.51) by $\Delta_t \Delta_g^3$. For the acoustics results presented here the scale factor is in the FDTD simulations.

The time evolution of the Yee FDTD acoustics algorithm and the Yee-dispersion-shifted theory are shown in Figure 5.1. For illustrative purposes in this figure only, the source pulse is delayed by ten time steps, and the theoretical solution is non-causal due to setting $H(t) \equiv 1$. Note that at step 11 (the endpoint of the width two Δ_t pulse, at approximately time 6) the Yee simulation and the dispersion shifted theory overlap. The overlap remains excellent throughout the remaining time steps of the 65536 step simulation. This demonstrates that the width two Δ_t pulse is the proper source function for the given theoretical problem.

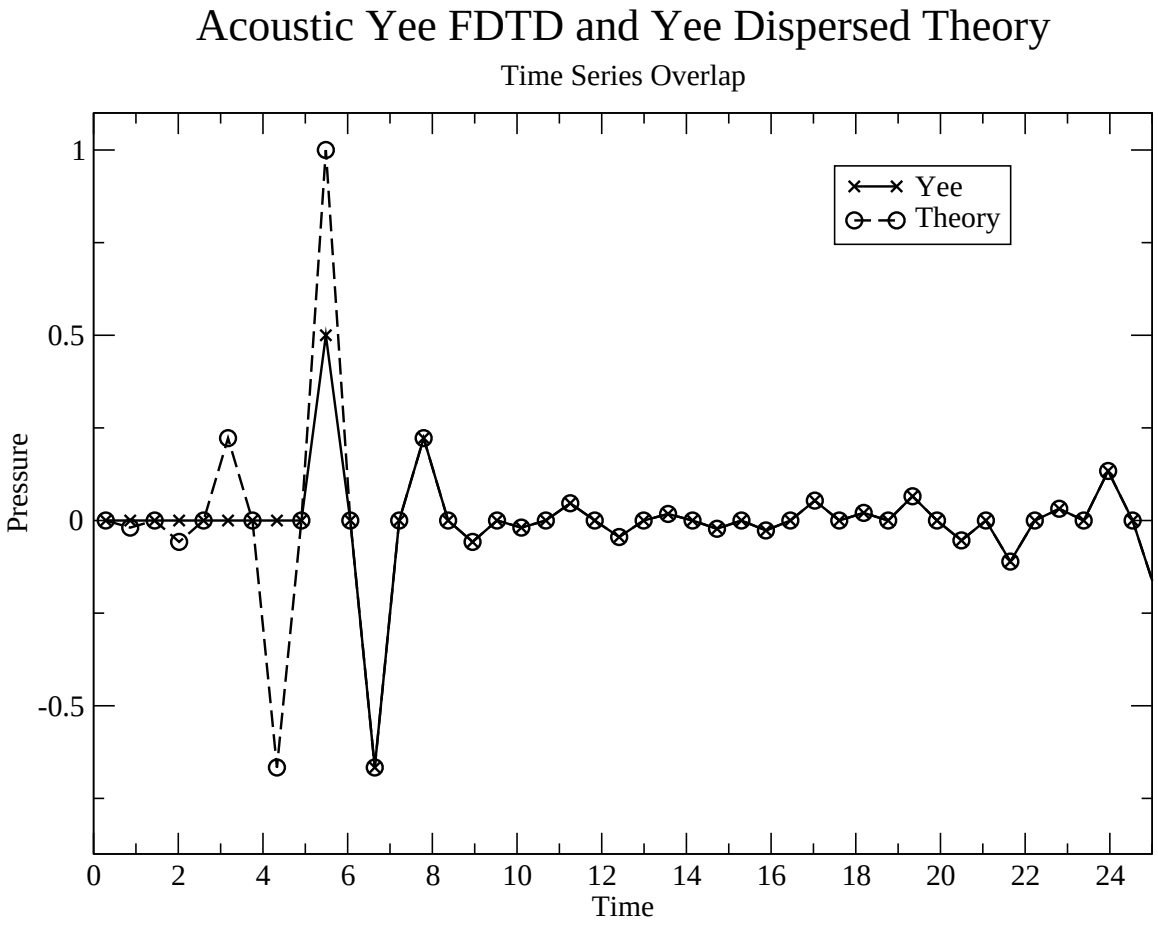

Figure 5.1: First time steps of a Yee FDTD simulation and Yee dispersion-shifted theoretical time evolution. The simulation source pulse starts at step 10 (about time 5). From time-step 11 (endpoint of the width $2\Delta_t$ pulse, about time 6) onward the theory and simulation overlap, and continue to do so for the remaining steps of the simulation. Given the resonator eigenfunctions and the Yee dispersion relation, the behavior of the Yee acoustics resonator is predictable.

5.4.4 Electromagnetics Resonator Analysis

As usual, the electromagnetics simulations (the computer programs) update the fields using the coupled first order system of Maxwell's curl equations, constitutive relations, boundary conditions, and initial conditions

$$\frac{\partial \mathbf{D}}{\partial t} = \boldsymbol{\nabla} \times \mathbf{H} - \mathbf{J}, \qquad \frac{\partial \mathbf{B}}{\partial t} = -\boldsymbol{\nabla} \times \mathbf{E}, \qquad \mathbf{r} \in \Omega\,,\ t \geq 0, \tag{5.52}$$

$$\mathbf{B} = \mu \mathbf{H}, \qquad \mathbf{D} = \epsilon \mathbf{E}, \qquad \mathbf{J}_c = \sigma \mathbf{E}, \qquad \mathbf{r} \in \Omega\,,\ t \geq 0, \tag{5.53}$$

$$\hat{\mathbf{n}} \times \mathbf{E} = 0, \qquad \mathbf{r} \in \partial\Omega\,,\ t \geq 0, \tag{5.54}$$

$$\mathbf{E} \equiv 0, \quad \mathbf{D} \equiv 0, \quad \mathbf{B} \equiv 0, \quad \mathbf{H} \equiv 0, \quad \mathbf{J} \equiv 0, \qquad \mathbf{r} \in \bar{\Omega}\,,\ t < 0. \tag{5.55}$$

For analysis the electromagnetics resonator is modeled as a boundary-value problem with a forcing function. The model for $\mathbf{E}(\mathbf{r}, t)$ is obtained by eliminating $\mathbf{H}, \mathbf{B}$, and $\mathbf{D}$ from (5.52)–(5.55), using the simplifying assumptions that the permittivity ϵ and permeability μ are invariant scalars, and the conductivity σ is zero. The resonator boundary value problem becomes

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E} + \epsilon\,\mu\,\frac{\partial^2 \mathbf{E}}{\partial t^2} = -\mu\,\frac{\partial \mathbf{J}}{\partial t}, \qquad \mathbf{r} \in \Omega\,,\ t \geq 0, \tag{5.56}$$

$$\hat{\mathbf{n}} \times \mathbf{E} = 0, \qquad \mathbf{r} \in \partial\Omega\,,\ t \geq 0, \tag{5.57}$$

$$\mathbf{E} \equiv 0, \qquad \mathbf{J} \equiv 0, \qquad \mathbf{r} \in \bar{\Omega}\,,\ t < 0. \tag{5.58}$$

The curl-curl operator is difficult to work with, so we rewrite it with a vector differential

identity

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E} = \boldsymbol{\nabla}\boldsymbol{\nabla} \cdot \mathbf{E} - \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\mathbf{E} = \boldsymbol{\nabla}\boldsymbol{\nabla} \cdot \mathbf{E} - \nabla^2 \mathbf{E}. \tag{5.59}$$

Now we obtain an expression for $\boldsymbol{\nabla} \cdot \mathbf{E}$ in terms of $\mathbf{J}$

$$\frac{\partial \mathbf{D}}{\partial t} = \boldsymbol{\nabla} \times \mathbf{H} - \mathbf{J} \implies \epsilon \boldsymbol{\nabla} \cdot \frac{\partial \mathbf{E}}{\partial t} = -\boldsymbol{\nabla} \cdot \mathbf{J} \Longleftrightarrow \boldsymbol{\nabla} \cdot \mathbf{E} = -\frac{1}{\epsilon} \int_{0^-}^{t} \boldsymbol{\nabla} \cdot \mathbf{J} \, \mathrm{d}\tau. \tag{5.60}$$

Thus, without approximation in the homogeneous region, the $\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E}$ can be written

$$\boldsymbol{\nabla} \times \boldsymbol{\nabla} \times \mathbf{E} = -\nabla^2 \mathbf{E} - \frac{1}{\epsilon} \boldsymbol{\nabla} \int^{t} \boldsymbol{\nabla} \cdot \mathbf{J} \, \mathrm{d}\tau \tag{5.61}$$

The problem is then

$$-\nabla^2 \mathbf{E} + \epsilon \, \mu \, \frac{\partial^2 \mathbf{E}}{\partial t^2} = -\mu \, \frac{\partial \mathbf{J}}{\partial t} + \frac{1}{\epsilon} \boldsymbol{\nabla} \int_{0^-}^{t} \boldsymbol{\nabla} \cdot \mathbf{J} \, \mathrm{d}\tau, \qquad \mathbf{r} \in \Omega \, , \, t \geq 0, \tag{5.62}$$

$$\boldsymbol{\nabla} \cdot \mathbf{E} = -\frac{1}{\epsilon} \int_{0^-}^{t} \boldsymbol{\nabla} \cdot \mathbf{J} \, \mathrm{d}\tau, \qquad \mathbf{r} \in \Omega \, , \, t \geq 0, \tag{5.63}$$

$$\hat{\mathbf{n}} \times \mathbf{E} = 0, \qquad \mathbf{r} \in \partial\Omega \, , \, t \geq 0, \tag{5.64}$$

$$\mathbf{E} \equiv 0, \qquad \mathbf{J} \equiv 0, \qquad \mathbf{r} \in \bar{\Omega} \, , \, t < 0, \tag{5.65}$$

Equation (5.62) is the governing PDE, (5.64) gives the boundary conditions, and (5.65) are causality conditions. Note that the current density, $\mathbf{J}$, is the current in (5.52), it can be data, as here, or it can depend on the $\mathbf{E}$ field.

The formal solution for the $\mathbf{E}$ field is

$$\mathbf{E}(\mathbf{r},t) = \sum_{m,p,q}^{\infty} \left[A^X_{mpq}(t)\Phi^X_{mpq}(\mathbf{r})\hat{\mathbf{x}} + A^Y_{mpq}(t)\Phi^Y_{mpq}(\mathbf{r})\hat{\mathbf{y}} + A^Z_{mpq}(t)\Phi^Z_{mpq}(\mathbf{r})\hat{\mathbf{z}} \right], \quad (5.66)$$

where $A^X_{mpq}(t)$ is the x axis mode time dependence, and where $\Phi^X_{mpq}(\mathbf{r})$ is the x axis eigenfunction, and similarly for the y and z axes. Take the current density to be a centered, two cell long z-directed filament

$$\mathbf{J}(\mathbf{r},t) = \delta(x)\delta(y)(H(z+\Delta_g) - H(z-\Delta_g))f(t)\hat{\mathbf{z}}, \quad (5.67)$$

where $\delta()$ is a delta function, $H()$ is a step function, and $f(t)$ is the time dependence of the current density. Now the symmetry of the centered source, and the boundary conditions (5.64) can be used to find the eigenfunctions. The symmetry and boundary conditions require that the m, p indexes are odd, and the q index is even. For notational simplicity let $k_m = m\pi/L$, $k_p = p\pi/L$ and $k_q = q\pi/L$. Then the eigenfunctions, satisfying the boundary condition (5.64), are

$$\begin{aligned} \Phi^X_{mpq}(\mathbf{r}) &= \sin(k_m x)\cos(k_p y)\sin(k_q z), \\ \Phi^Y_{mpq}(\mathbf{r}) &= \cos(k_m x)\sin(k_p y)\sin(k_q z), \\ \Phi^Z_{mpq}(\mathbf{r}) &= \cos(k_m x)\cos(k_p y)\cos(k_q z). \end{aligned} \quad (5.68)$$

Expand the forcing function (the right-hand-side of (5.62)) in the resonator eigenfunctions

$$\begin{aligned}\mathbf{F}(\mathbf{r},t) &= -\mu\frac{\partial \mathbf{J}}{\partial t} + \frac{1}{\epsilon}\boldsymbol{\nabla}\int_{0^-}^{t}\boldsymbol{\nabla}\cdot\mathbf{J}\,\mathrm{d}\tau \\ &= \sum_{\substack{m,p=1\\ \text{ODD}}}^{\infty}\sum_{\substack{q=0\\ \text{EVEN}}}^{\infty}\left[B^X_{mpq}(t)\Phi^X_{mpq}(\mathbf{r})\hat{\mathbf{x}} + B^Y_{mpq}(t)\Phi^Y_{mpq}(\mathbf{r})\hat{\mathbf{y}} + B^Z_{mpq}(t)\Phi^Z_{mpq}(\mathbf{r})\hat{\mathbf{z}}\right].\end{aligned} \tag{5.69}$$

Then the B coefficients are found in the usual way, yielding

$$\begin{aligned}B^X_{mpq}(t) =& \frac{16}{L^3}\sin(\Delta_g k_q)\frac{k_m}{\epsilon}\int_{0^-}^{t} f(\tau)\,\mathrm{d}\tau, \\ B^Y_{mpq}(t) =& \frac{16}{L^3}\sin(\Delta_g k_q)\frac{k_p}{\epsilon}\int_{0^-}^{t} f(\tau)\,\mathrm{d}\tau, \\ B^Z_{mpq}(t) =& \begin{cases}\frac{-16}{L^3}\sin(\Delta_g k_q)\left(\frac{k_q}{\epsilon}\int_{0^-}^{t} f(\tau)\,\mathrm{d}\tau + \frac{\mu}{k_q}f'(t)\right), & q>0 \\ \frac{-8\Delta_g}{L^3}\mu f'(t), & q=0.\end{cases}\end{aligned} \tag{5.70}$$

Once again the series expansions, (5.66) and (5.69), are substituted into the governing PDE (5.62), giving auxiliary ODE's for each of the A coefficients.

$$\omega^2_{mpq}A_{mpq}(t) + A''_{mpq}(t) = c^2 B_{mpq}(t). \tag{5.71}$$

Choose the time dependence of the current density to be a doublet, $f(t) = \delta'(t)$. Then the

solutions to the auxiliary ODE's, satisfying (5.62)–(5.65), are

$$
\begin{aligned}
A^X_{mpq}(t) =& \frac{16c^2}{L^3}\sin(\Delta_g k_q)\frac{k_m}{\epsilon\,\omega_{mpq}}\sin(\omega_{mpq}t)H(t),\\
A^Y_{mpq}(t) =& \frac{16c^2}{L^3}\sin(\Delta_g k_q)\frac{k_p}{\epsilon\,\omega_{mpq}}\sin(\omega_{mpq}t)H(t),\\
A^Z_{mpq}(t) =& \begin{cases} \frac{-16c^2}{L^3}\sin(\Delta_g k_q)\left(\frac{\mu\,\delta(t)}{k_q}+\left(\frac{k_q}{\epsilon\,\omega_{mpq}}-\frac{\mu\,\omega_{mpq}}{k_q}\right)\sin(\omega_{mpq}t)H(t)\right), & q>0\\ \frac{-8c^2\Delta_g}{L^3}\left(\mu\,\delta(t)-\mu\,\omega_{mpq}\sin(\omega_{mpq}t)H(t)\right), & q=0, \end{cases}
\end{aligned}
\tag{5.72}
$$

where the mode frequencies ω_{mpq} are given by (5.49).

Because of the sampling considerations, the indexes must satisfy $m, p < M$ and $q < M-1$. Analytically, the potential between the sample nodes is given by a line integral of the **E** field, along the z axis sample line. The value of "potential" extracted from the simulations is the sum of the two central E_z field samples times the grid step size. The predicted approximate potential is then the truncated (band-limited) sum

$$
U(t) = -C_1 \sum_{\substack{m,p=1\\ \text{ODD}}}^{M-1}\sum_{\substack{q=0\\ \text{EVEN}}}^{M-2} A^Z_{mpq}(t)2\Delta_g\cos\left(\frac{q\pi}{2M}\right), \tag{5.73}
$$

where C_1 is a constant described below.

In the electromagnetics case the theoretical source function is a doublet $\delta'(t)$. The obvious finite doublet would be the time series $(\cdots, 0, 1, -1, 0, \cdots)$. Again, two samples per pulse are required. There are several possible ways to rewrite the obvious finite doublet to have two samples per pulse. The choice which seems to produce the results closest to the theory is obtained by spreading each polarity of the obvious doublet over two steps, as was

done with the acoustics finite delta. The time series is then $(\cdots, 0, 1/2, 0, -1/2, 0, \cdots)$. This spaced-doublet is used as the unit amplitude finite doublet in all the electromagnetic simulation results presented here.

The finite-impulse amplitude scaling issue also applies here, as in acoustics. In electromagnetics, the vector current density **J** has specific physical meaning, so the finite-impulse scale factors are multiplied into (5.73). Here we have a doublet signal, applied to a current density. The scale factor is then $C_1 = \Delta_t^2 \Delta_g^2$, this factor is multiplied into (5.73) for the results we present.

A plot of the last few time steps of the potential sample and prediction is shown in Figure 5.2. This plot shows that there is fair agreement between the Yee-dispersed theoretical prediction and the simulation.

5.4.5 Algorithm Numerical Implementation

In order to implement the algorithms presented here, some method of computing derivatives (5.4)–(5.6) or (5.82), which is presented in the next section, is required. Shannon's sampling and reconstruction methods are used here. Reconstruction methods can be described as follows. Given reconstruction basis functions ϕ, some field component f is exactly reconstructed at the arbitrary point (x, y, z) from the discrete samples $f(i, j, k)$ with the infinite sums

$$f(x,y,z) = \sum_i \sum_j \sum_k \phi\left(\frac{x}{\Delta_g} - i\right) \phi\left(\frac{y}{\Delta_g} - j\right) \phi\left(\frac{z}{\Delta_g} - k\right) f(i\Delta_g, j\Delta_g, k\Delta_g), \tag{5.74}$$

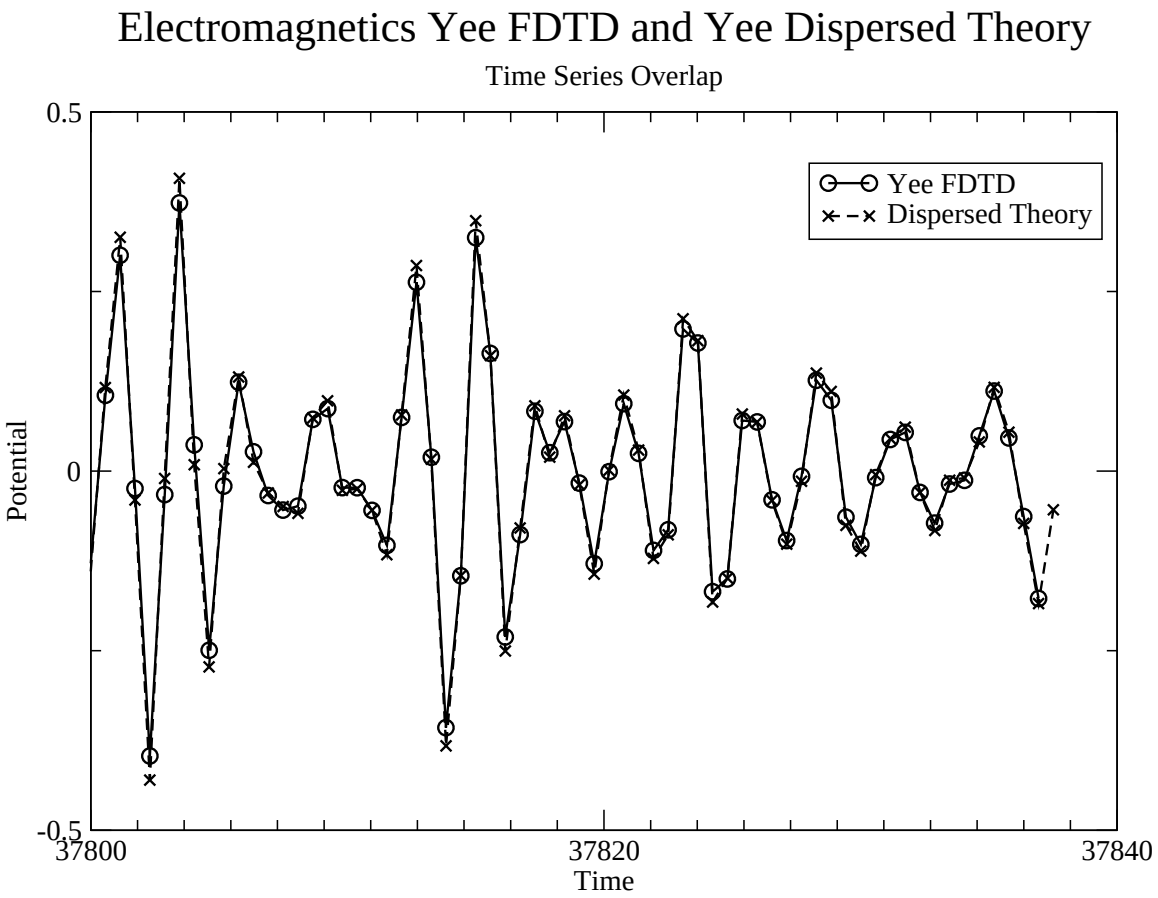

Figure 5.2: Plot of the last few time steps of the potential sample in an electromagnetic Yee FDTD resonator simulation and Yee dispersion-shifted theoretical time evolution of the resonator. The overlap here between theory and simulation is fair, though not excellent as in the acoustics case. The electromagnetic simulations are run with the spaced-doublet pulse.

where for Shannon reconstruction the basis functions ϕ are the $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$ functions. For notational simplicity let $g(x, y, z) = \phi(x)\phi(y)\phi(z) = g(\mathbf{r})$, and use vector notation for sum indexes, so (5.74) becomes

$$f(\mathbf{r}) = \sum_{\mathbf{I}} g\left(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I}\right) f(\Delta_g \mathbf{I}). \tag{5.75}$$

Now compute a spherical gradient, writing the function $f(\mathbf{r})$ with a reconstruction sum

$$\boldsymbol{\nabla}_{\mathrm{s}} f(\mathbf{r}) = \boldsymbol{\nabla}_{\mathrm{s}} \sum_{\mathbf{I}} g\left(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I}\right) f(\Delta_g \mathbf{I}) = \sum_{\mathbf{I}} f(\Delta_g \mathbf{I}) \boldsymbol{\nabla}_{\mathrm{s}} g\left(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I}\right), \tag{5.76}$$

where the exchange of the summation and $\boldsymbol{\nabla}_{\mathrm{s}}$ is permitted because the $\boldsymbol{\nabla}_{\mathrm{s}}$ only operates on $\mathbf{r}$, and not on $\mathbf{I}$. Additionally in an algorithm the sums are finite, so convergence issues do not apply. The function $\boldsymbol{\nabla}_{\mathrm{s}} g(\mathbf{r})$ can be precomputed. So we define the update coefficients

$$\mathbf{C}_G(\mathbf{r}) = \boldsymbol{\nabla}_{\mathrm{s}} g(\mathbf{r}) = \frac{\partial}{\partial V} \iint_{\partial \mathrm{B}} \hat{\mathbf{n}} g(\mathbf{r} + \mathbf{r}')\, \mathrm{d}s = \frac{1}{4\pi\rho^2} \frac{\partial}{\partial \rho} \iint_{\partial \mathrm{B}} \hat{\mathbf{n}} g(\mathbf{r} + \mathbf{r}')\, \mathrm{d}s, \tag{5.77}$$

where the surface integration is over the spherical surface of the operator. This is a complicated calculation which is performed numerically for a fixed radius. In terms of the

coefficients, the spherical gradient, divergence, and curl are

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi(\mathbf{r}) = \sum_{\mathbf{I}} \mathbf{C}_G \left(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I} \right) \psi(\Delta_g \mathbf{I}), \tag{5.78}$$

$$\boldsymbol{\nabla}_{\mathrm{s}} \cdot \mathbf{W}(\mathbf{r}) = \sum_{\mathbf{I}} \mathbf{C}_D \left(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I} \right) \cdot \mathbf{W}(\Delta_g \mathbf{I}), \tag{5.79}$$

$$\boldsymbol{\nabla}_{\mathrm{s}} \times \mathbf{W}(\mathbf{r}) = \sum_{\mathbf{I}} \mathbf{C}_C \left(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I} \right) \times \mathbf{W}(\Delta_g \mathbf{I}), \tag{5.80}$$

where $\mathbf{C}_G$, $\mathbf{C}_D$ and $\mathbf{C}_C$ are the gradient, divergence and curl update coefficients. Equations (5.78)–(5.80) are quite general. Depending on the contents of the coefficient arrays, they could provide classic Yee, higher-order Yee, Forgy [13], direct-domain equivalent of pseudo-spectral time-domain (PSTD) [32], or other algorithms. The small-cube acoustics algorithm (discussed in the next section) modeling system (5.31) uses (5.78) and (5.79) to compute the spatial derivatives. The spherical-derivative electromagnetics algorithm uses two staggered curls (5.80), to compute the partial derivatives. We need only compute the spatial derivatives of the fields for coordinates $\mathbf{r}$ on a discrete grid, typically limited to integer or half-integer multiples of the grid size Δ_g. Only one octant of one component of $\mathbf{C}_G$ needs to be computed, the other octants and components can be obtained from simple symmetry considerations. Additional symmetries could be used to further reduce the coefficient calculation. Similar considerations allow finding the $\mathbf{C}_D$ and $\mathbf{C}_C$ coefficients from the $\mathbf{C}_G$ if staggering is properly considered. For a collocated grid $\mathbf{C}_G = \mathbf{C}_D = \mathbf{C}_C$, in a staggered grid there are simple fixed shift offsets between the $\mathbf{C}$'s.

5.4.6 Small-Cube Operator Algorithm

In practice obtaining the update coefficients (5.77) is quite numerically expensive, nevertheless, the electromagnetics results presented here uses these. To more readily obtain coefficients a simplified approach we term the small-cube algorithm (where the "small-cube" refers to the shape of the operators and not to the resonator) is used in the acoustics results presented here. The small-cube operator algorithm simplifies the partial derivatives in the right-hand side of (5.4)–(5.6), which are approximated by the finite difference over a small cubic volume. This approximation greatly simplifies calculating the update coefficients used in numerical algorithms. For the gradient this simplification results in

$$\boldsymbol{\nabla}_{\mathrm{s}}\psi \approx \frac{\iint_{S(\Delta v)} \hat{\mathbf{n}}\psi \,\mathrm{d}s - \iint_{S(0)} \hat{\mathbf{n}}\psi \,\mathrm{d}s}{\Delta v} = \frac{1}{\Delta v}\iint_{S(\Delta v)} \hat{\mathbf{n}}\psi \,\mathrm{d}s = \boldsymbol{\nabla}_{\mathrm{c}}\psi, \tag{5.81}$$

where the integral over zero surface area, $S(0)$, yields zero, and $\boldsymbol{\nabla}_{\mathrm{c}}$ denotes the small-cube operator. The small-cube divergence and curl are defined in a similar manner, so the operators are

$$\begin{aligned}
\boldsymbol{\nabla}_{\mathrm{c}}\psi =& \frac{1}{\Delta v}\iint_{\partial \mathrm{v}} \hat{\mathbf{n}}\psi \,\mathrm{d}s \approx \boldsymbol{\nabla}_{\mathrm{s}}\psi, \\
\boldsymbol{\nabla}_{\mathrm{c}} \cdot \mathbf{W} =& \frac{1}{\Delta v}\iint_{\partial \mathrm{v}} \hat{\mathbf{n}} \cdot \mathbf{W} \,\mathrm{d}s \approx \boldsymbol{\nabla}_{\mathrm{s}} \cdot \mathbf{W}, \\
\boldsymbol{\nabla}_{\mathrm{c}} \times \mathbf{W} =& \frac{1}{\Delta v}\iint_{\partial \mathrm{v}} \hat{\mathbf{n}} \times \mathbf{W} \,\mathrm{d}s \approx \boldsymbol{\nabla}_{\mathrm{s}} \times \mathbf{W},
\end{aligned} \tag{5.82}$$

where Δv is the volume of the grid-aligned small-cube operator, with surface $\partial \mathrm{v}$. This approximation to the algorithm is equivalent, by integral theorems (5.1)–(5.3), to computing the volume average of the usual divergence, gradient or curl over the small cube and using

these averages as the algorithmic derivatives.

To find the effect of the small cube operator on scalar and vector plane waves the integrals in (5.82) are computed over a cube with sides of length $b\Delta_g$, yielding

$$\boldsymbol{\nabla}_{\mathrm{c}}\psi(\mathbf{r}) = i\,\mathrm{j}_0(K_x)\,\mathrm{j}_0(K_y)\,\mathrm{j}_0(K_z)\,\mathbf{k}\,\psi(\mathbf{r}), \tag{5.83}$$

$$\boldsymbol{\nabla}_{\mathrm{c}}\cdot\mathbf{A}(\mathbf{r}) = i\,\mathrm{j}_0(K_x)\,\mathrm{j}_0(K_y)\,\mathrm{j}_0(K_z)\,\mathbf{k}\cdot\mathbf{A}(\mathbf{r}), \tag{5.84}$$

$$\boldsymbol{\nabla}_{\mathrm{c}}\times\mathbf{A}(\mathbf{r}) = i\,\mathrm{j}_0(K_x)\,\mathrm{j}_0(K_y)\,\mathrm{j}_0(K_z)\,\mathbf{k}\times\mathbf{A}(\mathbf{r}), \tag{5.85}$$

where $K_x = b\Delta_g k_x/2$, $K_y = b\Delta_g k_y/2$, and $K_z = b\Delta_g k_z/2$. Writing the derivatives with spherical Bessel functions is a notational convenience, the natural results of the integrations are complex exponentials or trig functions. The dispersion equation for an acoustics system, (5.31), is then easily written using equations (5.83), (5.84) and (5.13). The result can be expressed as

$$\frac{4}{\Delta_t^2}\sin^2\left(\frac{\omega\Delta_t}{2}\right) = (ck)^2\,\mathrm{j}_0^2(K_x)\,\mathrm{j}_0^2(K_y)\,\mathrm{j}_0^2(K_z). \tag{5.86}$$

To find the stability limit s (where $s = c\Delta_t/\Delta_g$), (5.86) is solved for the angular frequency

$$\omega = \frac{2}{\Delta_t}\arcsin\left(s\frac{k\Delta_g}{2}\left|\,\mathrm{j}_0(K_x)\,\mathrm{j}_0(K_y)\,\mathrm{j}_0(K_z)\right|\right). \tag{5.87}$$

Stability requires that ω be real for all permitted $\mathbf{k}$'s, so the argument of the $\arcsin$ function

must be between zero and one. Thus the stability limit s as a function of b is given by

$$\frac{1}{s(b)} = \max_{\substack{\mathbf{k} \\ |k_x|,|k_y|,|k_z| \le \pi/\Delta_g}} \left(\frac{k\Delta_g}{2} \left| \mathrm{j}_0(K_x)\, \mathrm{j}_0(K_y)\, \mathrm{j}_0(K_z) \right| \right). \tag{5.88}$$

While the exact algorithm can be stabilized at any Δ_t, this approximation cannot. For arbitrary b the $\max$ in (5.88) is difficult to find. However, if the size of the operator is limited, the expression for the stability limit reduces to

$$s(b) = \frac{2}{\pi\sqrt{3}} \left| \frac{b\pi/2}{\sin b\pi/2} \right|^3 \qquad \text{where} \qquad 0 \le b \lesssim 0.615. \tag{5.89}$$

Equations (5.88) or (5.89) can be numerically solved for the size of the cubic operator for a given stability limit. To obtain a stability limit of $1/\sqrt{3}$, the Yee stability limit, the cube edge length is set to $b \approx 0.5958636$. For the small-cube operator the coefficient field $\mathbf{C}$'s are products of three independent 1D functions, so (in principle) the full 3D coefficient arrays do not need to be stored. The small-cube coefficients are

$$\mathbf{C}(\mathbf{r}) = \boldsymbol{\nabla}_{\mathrm{c}} g(\mathbf{r}) = \frac{1}{\Delta v} \iint_{\partial \mathrm{v}} \hat{\mathbf{n}} g(\mathbf{r} + \mathbf{r}')\, \mathrm{d}s \approx \boldsymbol{\nabla}_{\mathrm{s}} g(\mathbf{r}). \tag{5.90}$$

Performing the surface integrations over the six faces of the cube operator (5.90) yields

$$\begin{aligned} \mathbf{C}(\mathbf{r}) = \frac{1}{b^3} \big[& \big(\phi(x^+) - \phi(x^-)\big) \big(\Phi(y^+) - \Phi(y^-)\big) \big(\Phi(z^+) - \Phi(z^-)\big)\, \hat{\mathbf{x}} \\ & + \big(\Phi(x^+) - \Phi(x^-)\big) \big(\phi(y^+) - \phi(y^-)\big) \big(\Phi(z^+) - \Phi(z^-)\big)\, \hat{\mathbf{y}} \\ & + \big(\Phi(x^+) - \Phi(x^-)\big) \big(\Phi(y^+) - \Phi(y^-)\big) \big(\phi(z^+) - \phi(z^-)\big)\, \hat{\mathbf{z}} \big], \end{aligned} \tag{5.91}$$

where $x^+ = x + b\Delta_g/2$ and $x^- = x - b\Delta_g/2$, etc., and

$$\Phi(x) = \int_0^x \phi(u)\,\mathrm{d}u = \int_0^x \operatorname{sinc} u\,\mathrm{d}u = \frac{\operatorname{Si}(\pi x)}{\pi}, \tag{5.92}$$

is the "sine integral" function. These coefficients perform qualitatively similarly to the exact coefficients, but are much simpler to obtain. In effect, the only difference between the "exact" and "small-cube" algorithms is the coefficients used in the update derivatives (5.78)–(5.80).

5.4.7 Numerical Results

Figure 5.3 shows the acoustics calculations. The plots show the magnitude of the FFT vs. frequency of the sample point data. Top plot is the theory, middle plot is Yee algorithm and bottom plot is cubic derivative algorithm. Note that in the Yee algorithm fourth and fifth spectral lines exhibit mode-splitting [33, 17], due to the anisotropy of the Yee algorithm. These two lines are distinct even though they correspond to degenerate modes in the continuous world and hence should appear as a single resonant line. Other lines also show splitting. The bottom plot in Figure 5.3 shows the acoustics simulation results for an identically configurated resonator, but using the small-cube divergence and gradient operators. Note again that the fourth spectral line has two closely-spaced peaks, this line is split due to anisotropy, though the splitting is much smaller than in the Yee algorithm, the splitting of the fourth line is too small to observe on this plot. The mode structure for the small-cube algorithm is much closer to the theoretical mode structure than the Yee algorithm.

Figure 5.4 shows the results of the calculations for the electromagnetic resonator. The

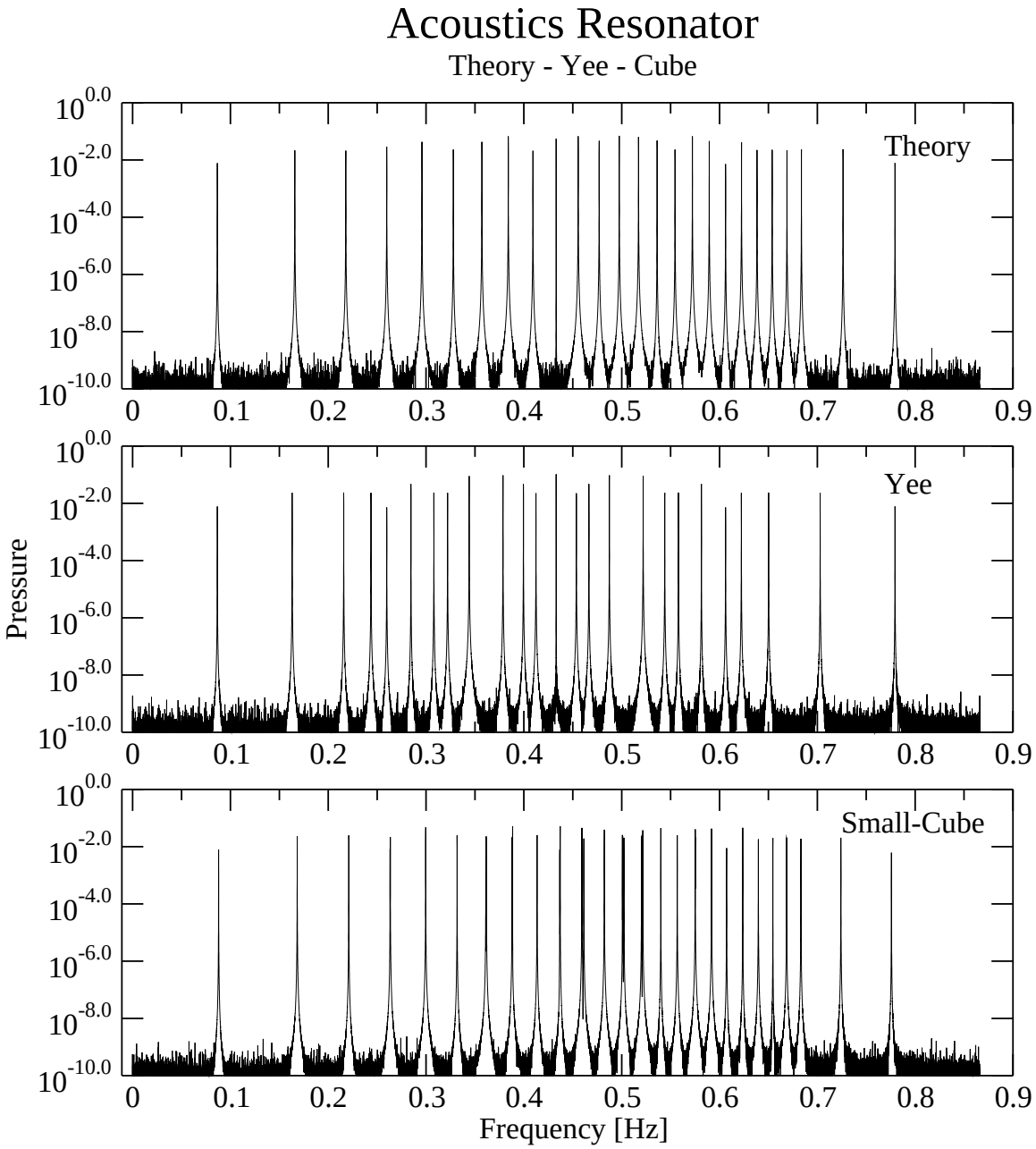

Figure 5.3: Spectra of solutions to the acoustics resonator problem. Top plot is the theory (5.51), middle plot is the Yee algorithm and bottom plot is the small-cube algorithm. The anisotropy of the Yee algorithm causes mode splitting, combining, and shuffling, thus scrambling the modal structure. The small-cube algorithm does exhibit some splitting, which is visible in some lines.

plots show the magnitude of the FFT vs. frequency of the sample data. The top plot shows the theoretical prediction given by (5.73). The middle plot is the result of a Yee algorithm simulation and the bottom plot is the spherical derivatives algorithm. The electromagnetics Yee algorithm is divergence free [7], unless the simulation problem has required divergences, e.g., the current deposits charge [12]. The source current has no DC component, and no DC is visible. The mode structure in the spherical algorithm is much better than the Yee Algorithm, but at higher frequencies splitting is clearly visible.

5.4.8 Theoretical vs. Numerical Dispersion

The theoretical spherical operator dispersion relation, (5.34) or (5.39), repeated here for convenience, is

$$\left(\frac{2}{c\Delta_t}\right)^2 \sin^2\left(\frac{\omega\Delta_t}{2}\right) = \frac{1}{(a\Delta_g)^2}\sin^2\left(ka\Delta_g\right). \tag{5.93}$$

The results for the spherical operator shown in Figure 5.4 are not exact, as expected from analysis of the algorithm when run according to the exactness condition (5.27). In practice the summations in (5.78)–(5.80) need to be carried out over the finite-sized model space, hence the reconstruction-based computation is an approximation of the algorithm. This means that the dispersion relation above only approximately characterizes a numerical implementation of the algorithm. The true numerical dispersion relation is then given in terms of finite sums. In acoustics for example

$$\left(\frac{2}{c\Delta_t}\right)^2 \sin^2\left(\frac{\omega\Delta_t}{2}\right) = e^{-i(\mathbf{k}\cdot\mathbf{r})}\sum_{\mathbf{I}} \mathbf{C}_D(\frac{\mathbf{r}}{\Delta_g} - \mathbf{I}) \cdot \sum_{\mathbf{J}} \mathbf{C}_G(\mathbf{I}-\mathbf{J})e^{i\mathbf{k}\cdot\Delta_g\mathbf{J}}. \tag{5.94}$$

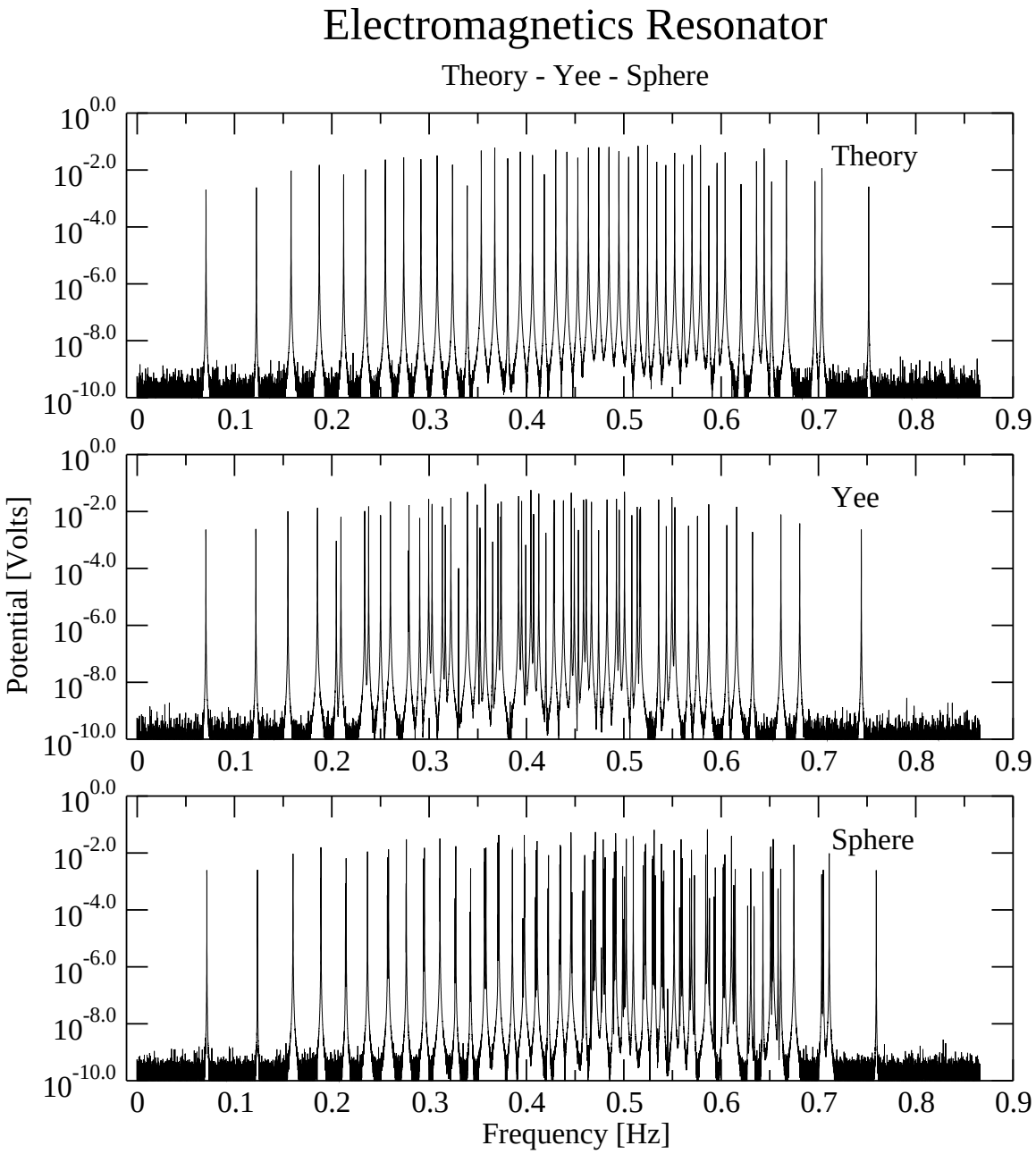

Figure 5.4: Spectra of solutions to the electromagnetics resonator problem. Top plot is the theory (5.73), middle plot is Yee algorithm and bottom plot is spherical derivative algorithm. Again note the mode structure is scrambled in the Yee algorithm. The spherical algorithm shows much less mode splitting, the mode structure is closer to the ideal.

This expression has a huge number of terms ($> 10^6$ complex exponentials), even for these small resonators.

5.5 Chapter Summary

The spherical operators presented here provide for theoretically exact 3D time-domain differential equation solvers. The theory of exact 3D time-domain solvers can be used to guide the construction of practical algorithms. We have shown the theoretical existence of 3D time-domain solvers which can, in principle, be exact for any time-step size. While the volume algorithms provides very good performance to the sampling limit, the $O[n^2]$ per axis cost is far too computationally expensive to be practical for much larger model spaces than the small resonators used here. Nonetheless, we have a theoretical framework for exact algorithms with a "magic" time step in 3D, which can be used to guide the construction of practical algorithms.

A well known method to improve performance is to use fast transform techniques. Such methods would still use the full space, i.e., the operators would remain global. Alternatively, preliminary testing suggests that practical algorithms can be made with some set of only local neighbor nodes contributing to the operators. Such local operators are much less computationally expensive than global reconstruction. The range of the operator could be increased or decreased to improve accuracy or speed. A method to reduce the needed computational range of the volume operators is to use reconstructors with faster spatial falloff than the Shannon reconstructors. This can be accomplished by modifying the rectangular spectrum of the Shannon reconstructor to have a slower falloff in the spec-

trum of the reconstructor—since a slower spectral falloff will produce a faster falloff in the reconstructor. The purpose of a localized reconstructor is to retain the same algorithmic cost as the Yee algorithm, albeit with a much larger constant cost factor.

The test implementations show potential of the method, but an efficient implementation that exploits this theory has not yet been obtained. The tests demonstrate solvers for both acoustics and Maxwell's equations, with good results.

Chapter 6

Conclusions

For the electromagnetic resonators studied in Chapter 3, the dispersion relation accurately predicts the frequencies at which a rectangular resonator will resonate. Anisotropic dispersion of the Yee algorithm can split or combine modes. Furthermore, the dispersion shift can change the resonant frequencies so that a list of observed modes ordered by resonant frequency may or may not correspond in order to a list obtained from the continuum. Modes can be combined, split, and shuffled—resulting in a scrambled mode structure.

The constructed divergence FDTD algorithm presented in Chapter 4 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. Such a correction would shift peaks but would not undo any splitting or combining. The constructed algorithm was shown to yield good results for canonical scattering problems.

The spherical operators presented in Chapter 5 provide for theoretically exact 3D time-domain differential equation solvers for hyperbolic systems of coupled first order equations in an unbounded homogeneous space. We have shown the theoretical existence of 3D time-domain solvers which can, in principle, be exact for any time-step size—a “magic” time

stepping in 3D.

As used here the spherical operators were used on a cubic grid. The definition for the operators (5.4)–(5.6), does not reference any grid—the operators could be defined on a non-cubic grid. While the volume algorithms provides very good performance to the sampling limit, the algorithm, as implemented here, is computationally expensive.

Bibliography

[1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, pp. 302–307, Mar. 1966.

[2] 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*, vol. 10, no. 6, pp. 547–582, 1988.

[3] K. L. Shlager and J. B. Schneider, "Comparison of the dispersion properties of several low-dispersion finite-difference time-domain algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 51, pp. 642–653, Mar. 2003.

[4] J. von Neumann and R. D. Richtmyer, "On the numerical solutions of partial differential equations of parabolic type," in *John von Neumann Collected Works* (A. H. Taub, ed.), vol. V, pp. 652–663, Pergamon Press, 1963. Los Alamos Report LA-657, December 25, 1947.

[5] J. B. Schneider and C. L. Wagner, "FDTD dispersion revisited: Faster-than-light propagation," *IEEE Microwave Guided Wave Letters*, vol. 9, pp. 54–56, Feb. 1999.

[6] A. Taflove and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-23, no. 8, pp. 623–630, 1975.

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

[8] S. Dey and R. Mittra, "A locally conformal finite-difference time-domain (FDTD) algorithm for modeling three-dimensional perfectly conducting objects," *IEEE Microwave Guided Wave Letters*, vol. 7, pp. 273–275, Sept. 1997.

[9] C. J. Railton and J. B. Schneider, "An analytical and numerical analysis of several locally conformal FDTD schemes," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, pp. 56–66, Jan. 1999.

[10] J. L. Young, "The design of high-order, leap-frog integrators for Maxwell's equations," in *IEEE Antennas and Propagat. Soc. Int. Symposium*, vol. 1, (Orlando, FL), pp. 176–179, July 1999.

[11] J. B. Schneider and R. J. Kruhlak, "Dispersion of homogeneous and inhomogeneous waves in the Yee finite-difference time-domain grid," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, pp. 280–287, Feb. 2001.

[12] C. L. Wagner and J. B. Schneider, "Divergent fields, charge, and capacitance in FDTD simulations," *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, pp. 2131–2136, Dec. 1998.

[13] E. A. Forgy and W. C. Chew, "A time-domain method with isotropic dispersion and increased stability on an overlapped lattice," *IEEE Transactions on Antennas and Propagation*, vol. 50, pp. 983–996, July 2002.

[14] J. W. Nehrbass, J. O. Jevtić, and R. Lee, "Reducing the phase error for finite-difference methods without increasing the order," *IEEE Transactions on Antennas and Propagation*, vol. 46, pp. 1194–1201, Aug. 1998.

[15] 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*, vol. 4, (Atlanta, GA), pp. 1812–1815, June 1998.

[16] E. A. Forgy, "A time-domain method for computational electromagnetics with isotropic numerical dispersion on an overlapped latice," Master's thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1998.

[17] C. L. Wagner and J. B. Schneider, "On the analysis of resonators using finite-difference time-domain techniques," *IEEE Transactions on Antennas and Propagation*, vol. 51, pp. 2885–2890, Oct. 2003.

[18] J. B. Schneider, C. L. Wagner, and R. J. Kruhlak, "Simple conformal methods for finite-difference time-domain modeling of pressure-release surfaces," *Journal of the Acoustical Society of America*, vol. 104, pp. 3219–3226, Dec. 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 Transactions on Nuclear Science*, vol. 27, pp. 1829–1833, Dec. 1980.

[20] A. Taflove and K. Umashankar, "Radar cross section of general three-dimensional scatterers," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-25, pp. 433–440, Nov. 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," *Journal of Computational Physics*, vol. 114, no. 1, pp. 185–200, 1994.

[23] J. G. Maloney and K. E. Cummings, "Adaptation of FDTD techniques to acoustic modeling," in *11th Annual Review of Progress in Applied Computational Electromagnetics*, vol. 2, (Monterey, CA), pp. 724–731, Mar. 1995.

[24] 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 Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 44, pp. 816–822, July 1997.

[25] W. C. Chew and W. H. Weedon, "A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates," *Microwave and Optical Technology Letters*, vol. 7, pp. 599–604, Sept. 1994.

[26] H. Medwin and C. S. Clay, *Fundamentals of Acoustical Oceanography*. New York: Academic Press, 1998.

[27] S. Wolfram, *The Mathematica Book*. Cambridge, UK: Cambridge University Press, 4 ed., 1999.

[28] A. I. Boresenko and I. E. Tarapov, *Vector and Tensor Analysis with Applications*. Englewood Cliffs, N.J.: Prentice Hall, Revised English ed., 1968.

[29] J. A. Stratton, *Electromagnetic Theory*. New York: McGraw-Hill, 1941.

[30] M. Abramowitz and I. E. Stegun, *Handbook of Mathematical Functions*. Nat. Bur. Standards, 1985.

[31] H. De Raedt, K. Michielsen, J. S. Kole, and M. T. Figge, "Solving the maxwell equations by the Chebyshev method: A one-step finite-difference time-domain algorithm," *IEEE Transactions on Antennas and Propagation*, vol. 51, pp. 3155–3160, Nov. 2003.

[32] Q. H. Liu, "The pseudospectral time-domain (PSTD) method: A new algorithm for solutions of Maxwell's equations," in *IEEE Antennas and Propagat. Soc. Int. Symp.*, vol. 1, (Montréal, Canada), pp. 122–125, July 1997.

[33] C. L. Wagner and J. B. Schneider, "Using the dispersion relation to understand finite-difference time-domain worlds," in *International Conference on Electromagnetics in Advanced Applications (ICEAA 01)*, (Torino, Italy), pp. 375–378, Sept. 2001.