Mathematical modeling of metamaterials

Valjean Elizabeth Elander

University of Nevada, Las Vegas

Follow this and additional works at: http://digitalscholarship.unlv.edu/thesesdissertations

Part of the Applied Mathematics Commons, and the Electromagnetics and photonics Commons

Repository Citation
MATHEMATICAL MODELING OF METAMATERIALS

by

Valjean Elizabeth Elander

Bachelor of Science in Mathematics
University of Missouri, St. Louis
1999

Master of Science in Mathematics
University of Nevada, Las Vegas
2005

A dissertation submitted in partial fulfillment of
the requirements for the

Doctor of Philosophy in Mathematical Sciences
Department of Mathematical Sciences
College of Sciences

Graduate College
University of Nevada, Las Vegas
May 2011
THE GRADUATE COLLEGE

We recommend the dissertation prepared under our supervision by

Valjean Elizabeth Elander

entitled

Mathematical Modeling of Metamaterials

be accepted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mathematical Sciences

Jichun Li, Committee Chair
Michael Marcozzi, Committee Member
Monika Neda, Committee Member
Pengtao Sun, Committee Member
Hongtao Yang, Committee Member
Yiting Chen, Graduate Faculty Representative

Ronald Smith, Ph. D., Vice President for Research and Graduate Studies and Dean of the Graduate College

May 2011
Metamaterials are artificially structured nano materials with negative refraction index. The successful construction of such metamaterials in 2000 triggered a great interest in study of metamaterials by researchers from different areas. The discovery of metamaterials opened a wide potential for applications in diverse areas such as cloaking, sub-wavelength imaging, solar cell design and antennas.

In this thesis, we investigate the most popular Drude metamaterial model. More specifically, we first present a brief overview of metamaterials and their potential applications, then we discuss the well-posedness of this model, and develop several numerical schemes to solve it. We implement our schemes using MATLAB, and demonstrate their effectiveness through numerical simulations of the negative refraction and cloaking phenomena.
ACKNOWLEDGEMENTS

This is dedicated to my grandmother, who said to finish school, no matter what. And to my mother and father, who have been tremendously supportive of whatever I have chosen to do with my life, I give thanks. And, thanks to all my friends that have been there for me through thick and thin. And, of course, I wouldn’t have made it this far without the help and support of the professors at all of the academic institutions that I have been to, especially those on my committee at UNLV.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ABSTRACT</th>
<th>iii</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION TO METAMATERIALS</td>
<td>1</td>
</tr>
<tr>
<td>1.1 History of Metamaterials</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Basic Structures of Metamaterials</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Potential Applications of Metamaterials</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Outline of Dissertation</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 2 MODELING EQUATIONS AND METAMATERIAL SIMULATION</td>
<td>12</td>
</tr>
<tr>
<td>2.1 Governing Equations</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Basic Mathematical Properties of the Model</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Introduction to FDTD Method</td>
<td>18</td>
</tr>
<tr>
<td>2.4 The Yee Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>2.5 Numerical Dispersion and Stability</td>
<td>22</td>
</tr>
<tr>
<td>2.6 Absorbing Boundary Conditions</td>
<td>23</td>
</tr>
<tr>
<td>2.7 Dispersive FDTD Scheme for Metamaterial Simulation</td>
<td>24</td>
</tr>
<tr>
<td>2.8 Numerical Simulations</td>
<td>26</td>
</tr>
<tr>
<td>CHAPTER 3 SIMULATION OF CLOAKING PHENOMENON</td>
<td>38</td>
</tr>
<tr>
<td>3.1 Introduction to Cloaking</td>
<td>38</td>
</tr>
<tr>
<td>3.2 FDTD Method for Cloaking</td>
<td>40</td>
</tr>
<tr>
<td>3.3 Numerical Simulations</td>
<td>45</td>
</tr>
<tr>
<td>3.4 Example 1</td>
<td>45</td>
</tr>
<tr>
<td>3.5 Example 2</td>
<td>46</td>
</tr>
<tr>
<td>3.6 Example 3</td>
<td>48</td>
</tr>
<tr>
<td>CHAPTER 4 HIGH ORDER COMPACT SCHEMES FOR MAXWELL'S EQUATIONS</td>
<td>52</td>
</tr>
<tr>
<td>4.1 Introduction to Compact Difference Schemes</td>
<td>52</td>
</tr>
<tr>
<td>4.2 High Order Spatial Derivatives for Staggered Grids</td>
<td>53</td>
</tr>
<tr>
<td>4.3 High Order Temporal Derivatives for Staggered Grids</td>
<td>58</td>
</tr>
</tbody>
</table>
Table 4.1: \( L^\infty \) errors for solution \( \sin(\pi x) \cos(\pi y) \) . . . . . . . . . . . . . . . 61
Table 4.2: \( L^\infty \) errors for solution \( \exp(4xy) + 2x - 4xy \) . . . . . . . . . . . . . . 62
Table 4.3: \( L^\infty \) errors for \( CFL = 0.5 \) and \( \Delta x \) from 1/10 to 1/80 . . . . . . . . 65
Table 4.4: \( L^\infty \) errors for \( CFL = 0.1 \) and \( \Delta x \) from 1/10 to 1/80 . . . . . . . . 65
Table 4.5: \( L^\infty \) errors for \( CFL = 0.005 \) and \( \Delta x \) from 1/10 to 1/80 . . . . . . . . 65
Table 4.6: \( L^\infty \) errors for \( CFL = 0.5 \) and \( \Delta x \) from 1/40 to 1/320 . . . . . . . . 66
Table 4.7: \( L^\infty \) errors for \( CFL = 0.005 \) and \( \Delta x \) from 1/40 to 1/320 . . . . . . . . 66
Table 5.1: \( L^\infty \) errors with \( \tau = 10^{-5} \) for 10 time steps . . . . . . . . . . . . . . 74
Table 5.2: \( L^\infty \) errors with \( \tau = 10^{-6} \) for 100 time steps . . . . . . . . . . . . . . 74
Table 5.3: \( L^\infty \) errors with \( \tau = 10^{-7} \) for 1000 time steps . . . . . . . . . . . . . . 75
LIST OF FIGURES

Figure 1.1: Split ring resonator and unit cell example ............................... 2
Figure 1.2: A double negative metamaterial slab example .......................... 3
Figure 1.3: Electromagnetic spectrum ......................................................... 4
Figure 1.4: Holey metamaterial structure ..................................................... 4
Figure 1.5: Fishnet metamaterial structure .................................................. 5
Figure 1.6: Typical antenna field regions .................................................... 6
Figure 1.7: Gradient index lens ................................................................. 6
Figure 1.8: Communications satellite lined with metamaterial ......................... 7
Figure 1.9: Metamaterial lining of potential communications satellite ............... 8
Figure 1.10: Swiss roll metamaterial .......................................................... 8
Figure 1.11: Metamaterial antenna ............................................................. 9
Figure 1.12: Example of a tunable metamaterial ......................................... 10
Figure 2.1: Yee Lattice .............................................................................. 20
Figure 2.2: Single slab metamaterial with $n \approx -1$ .................................. 28
Figure 2.3: Single slab metamaterial with $n \approx -6$ .................................. 29
Figure 2.4: Metamaterial waveguide with rectangular slabs ......................... 31
Figure 2.5: Cylindrical metamaterial of radius $0.25 \lambda_0$ with $n \approx -1$ ......... 32
Figure 2.6: Cylindrical metamaterial of radius $0.5 \lambda_0$ with $n \approx -1$ .......... 33
Figure 2.7: Cylindrical metamaterial of radius $\lambda_0$ with $n \approx -1$ ............ 34
Figure 2.8: Cylindrical metamaterial of radius $0.25 \lambda_0$ with $n \approx -6$ ......... 35
Figure 2.9: Cylindrical metamaterial of radius $0.5 \lambda_0$ with $n \approx -6$ .......... 36
Figure 2.10: Cylindrical metamaterial of radius $\lambda_0$ with $n \approx -6$ ............ 37
Figure 3.1: Ray trajectories of waves through a cylindrical metamaterial cloak 38
Figure 3.2: First realized cloak .................................................................. 41
Figure 3.3: First realized cloak experimental and simulated results ................ 41
Figure 3.4: A MetaFlex nanoantenna geometry ......................................... 42
Figure 3.5: A MetaFlex metallic fishnet nanostructure ............................... 42
Figure 3.6: Example 1 of cloak simulations ................................................. 46
Figure 3.7: Example 2 of cloak simulations $E_x$ field ............................... 47
Figure 3.8: Example 2 of cloak simulations $H_z$ field ............................... 48
Figure 3.9: Example 3 of cloak simulations $E_x$ field ............................... 50
Figure 3.10: Example 3 of cloak simulations $H_z$ field ............................. 51
Figure 4.1: 1-D staggered grid ................................................................... 54
Figure 4.2: 2-D staggered grid used with $TE_z$ mode ............................... 63
Figure 5.1: Example of a triangular element used in DG method ................ 68
Figure 5.2: Relationship of elements used in DG method ........................... 69
CHAPTER 1

INTRODUCTION TO METAMATERIALS

1.1 History of Metamaterials

Metamaterials are types of man-made materials whose permittivity and permeability can be simultaneously negative for some common excitation wave frequency. Hence, metamaterials have a negative refraction index. Also, people call such materials double-negative (DNG) or negative index metamaterials (NIMs). They were discussed theoretically as early as 1967 by Victor Veselago, who published his paper on the subject in Russian titled *The Electrodynamics of Substances with Simultaneously Negative Values of $\epsilon$ and $\mu*$ \[65\], translated into English in 1968. This was the first time it was shown that refractive index could theoretically be negative. However, at that time, the equipment needed to fabricate such materials did not exist. People have been trying to create metamaterials since the end of World War II. It wasn’t until 33 years later when a composite material with negative refractive index was realized by a group of physicists at UC San Diego, lead by David Smith \[55\]. David Smith and Sir John Pendry realized how to physically create metamaterials only recently. This opened the doors to a new field of research. Sir John Pendry proved in 1999 that a three dimensional network of thin wires had negative permittivity \[46\]. Shortly after that, a network of copper split ring resonators (SRRs) was created that produced an effective negative permeability. Then, in 2000, Smith et. al. successfully combined the two to create the first double-negative metamaterial for a band of frequencies in the GHz range.
1.2 Basic Structures of Metamaterials

There are many different types of metamaterials. Terahertz, photonic, tunable, frequency selective surface (FSS) based, nonlinear, absorber, superlens, cloaking device, antenna, acoustic, and even seismic. Usually, they are periodic structures which incorporate a system of SRRs and wires. Pendry was the first person to be able to prove the usefulness of a functioning metamaterial in the 1990s. It so happens that this milestone was a metamaterial with a negative index of refraction in the microwave range. The SRR and the first realized unit cell of a DNG metamaterial created by Dr. Smith and his group at UC San Diego are shown in Figure 1.1. Figure 1.2 shows a periodic structure of these cells, which we refer to as a metamaterial slab.

![Figure 1.1](image)

**Figure 1.1.** (a) Example of a single square split-ring resonator, c = 0.25 mm, d = 0.30 mm, g = 0.46 mm, w = 2.62 mm. (b) A double negative unit cell. [54]

A split ring resonator (SRR) is one component of negative index materials. They are used extensively in many metamaterials, and are created by using a nonmagnetic metal such as copper. Their shape can be concentric squares, circles, and gapped
Figure 1.2. A double negative metamaterial lens example. Courtesy of NASA Glenn Research Center.

as necessary. Shortly after the production of the first DNG metamaterial, Sir John Pendry began working on another metamaterial that originally appeared in Veselago’s paper back in 1968: the perfect lens. This is a DNG metamaterial lens that can exceed the diffraction limit, a limitation of conventional lenses. In 2000, Pendry published a paper theorizing the use of a slab of negative refractive index material instead of a conventional lens and that such super lenses can be realized in the microwave band [47]. A reminder of the electromagnetic spectrum is given in Figure 1.3.

Another structure besides those using SRRs is a ”holey” metamaterial, shown in Figure 1.4. Holey metamaterial is capable of acoustic imaging at very deep-subwavelength scales [78]. Applications of which are discussed in the following section.

A fishnet structure is another type of metamaterial structure. It is a 3-D optical metamaterial with low loss. An example of this type of structure can be seen in Figure 1.5.
1.3 Potential Applications of Metamaterials

As we can see, the development of metamaterials is quite recent, and new developments are exploding. Although some of Pendry’s work was found to be unrealistic...
using double negative metamaterials, he was correct that both the propagating and evanescent waves are focused within the material, and that they converge at a focal point within the material and once again beyond. Scientists have continued studying lenses using metamaterials. In 2005, a silver metamaterial super lens was achieved to reconstruct images in the near field, less than one wavelength from the source. Field regions are provided in Figure 1.6. Negative index material was used in 2005 to create a gradient index (GRIN) lens which has the material permittivity and permeability matched closer to free space [23]. A GRIN lens that was created at the end of 2009 at Duke is shown in Figure 1.7.

A GRIN lens is made using a metamaterial created with a fiberglass material, that of which circuit boards are made of, and copper. Scientists are now trying to create a far field super lens using metamaterial. Currently, they can use a metamaterial lens to magnify both the propagating and evanescent waves, to make them large enough as
to be detectable by a conventional lens. Also, as recently as 2010, metamaterial was used to create a three-dimensional nanolens that achieves super-resolution imaging.
In January of this year, 2011, Shana Leonard wrote an article for the Medical Product Manufacturing News discussing the possible improvement of ultrasounds using holey metamaterial [31]. An example of holey metamaterial is shown in Figure 1.4.

Also in January of this year, an article in Penn State Live speaks of potentially using metamaterials for creating communications satellites, since this would save money by making them lighter and more energy-efficient, thus cheaper to make and send into space [43]. A 3-D rendering of a potential satellite is shown in Figure 1.8 where the lining of the satellite is made of metamaterial shown in Figure 1.9.

![Figure 1.8](image)

**Figure 1.8.** A 3-D rendering of a potential communications satellite lined with metamaterial. Courtesy of Penn State University

Since metamaterials oftentimes consist of creating a magnetic material using non-magnetic material components, its use in magnetic resonance imaging (MRI) is highly possible. This was written about in several articles, including [56] [18]. In Fountain’s
article, he quotes Dr. Eleftheriades, a professor at the University of Toronto, who says "A metamaterial coil could improve the signal-to-noise ratio in these machines, and improve the contrast of images". In the Metamaterials Review article [56], the Swiss roll was claimed to be the most suited to MRI applications. The Swiss roll is shown in Figure 1.10.

![Swiss roll metamaterial](image)

**Figure 1.10.** (A) A single Swiss roll metamaterial, 1 cm in diameter. (B) An array of Swiss roll elements (shown by the red circles), and the RF magnetic field from an M-shaped antenna. (C) MRI machine image, showing the field trough the slab.

Also in Fountain’s article, other applications mentioned are solar energy, cloaking
devices, cell phone antennas, and antennas molded to match the aerodynamic shape of a plane.

Many cell phones already contain metamaterial antennas that can handle multiple frequencies. One such antenna is pictured in Figure 1.11.

![Figure 1.11. Metamaterial antenna for cell phone usage. Courtesy of C. Holloway/National Institute of Standards and Technology.](image)

Metamaterials can be made to be nearly perfect absorbers of light. In this case, they may be useful in producing solar energy. The more photons it can absorb, the more energy can be produced.

It is hopeful that metamaterials can be applied to remote aerospace applications, sensor detection, infrastructure monitoring, smart solar power management, public safety, radomes, high-frequency battlefield communication, improving ultrasonic sensors, and shielding structures from earthquakes.

One type of metamaterial mentioned earlier is a tunable metamaterial. This is one that can be "tuned" to respond to different frequencies instead of only fixed
frequencies. Figure 1.12 shows one such development. Several applications can arise simply from this type of metamaterial alone, such as photovoltaics, goggles to block glare, and optical signal processing devices to speed communications.

Figure 1.12. An array of SRRs made of silver printed on a stretchy polymer. Courtesy of Atwater group, Caltech

It should be quite clear to the reader how important the discovery of metamaterials has been to mankind, and how useful they will serve to be in the future.

1.4 Outline of Dissertation

Following the introduction, Chapter 2 discusses the governing equations, along with background information of the finite-difference time-domain (FDTD) method. The governing equations for DNG metamaterials will be provided and explained, along with definitions of commonly used parameters pertaining to electrical engineer-
ing, physics, and metamaterial simulations. Our new dispersive FDTD scheme is introduced and explained here, and several different results are shown.

In Chapter 3, we start by giving a brief background of cloaking. After the introduction, the FDTD schemes are discussed, along with any new parameters and coefficients that differ from those used in Chapter 2. Results of simulations are provided.

Chapter 4 looks at some new high-order compact finite difference schemes that can be used to increase the speed and accuracy of the FDTD simulations of metamaterials.

We discuss the discontinuous Galerkin method in Chapter 5, and how it can be used for metamaterial simulation.

And, finally, we discuss major conclusions and directions for future work in Chapter 6.
CHAPTER 2
MODELING EQUATIONS AND METAMATERIAL SIMULATION

2.1 Governing Equations

We begin with the two equations that form the basis of Maxwell’s equations. The following are Faraday’s Law and Ampere’s Law, respectively, in a charge and current free medium:

\[
\begin{align*}
\frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \mathbf{E}, \\
\frac{\partial \mathbf{D}}{\partial t} &= \nabla \times \mathbf{H},
\end{align*}
\]

(2.1.1) (2.1.2)

where \(\mathbf{E}(\mathbf{x}, t)\) and \(\mathbf{H}(\mathbf{x}, t)\) are the electric and magnetic fields, respectively, and \(\mathbf{D}(\mathbf{x}, t)\) and \(\mathbf{B}(\mathbf{x}, t)\) are the electric and magnetic flux densities, respectively.

Since the above equations have more unknowns than number of equations, we need to use constitutive equations to make the problem well-posed. For a general medium, we need the following constitutive equations that relate the medium properties to the electric and magnetic fields:

\[
\begin{align*}
\mathbf{D} &= \varepsilon \mathbf{E} = \varepsilon_r \varepsilon_0 \mathbf{E} \\
\mathbf{B} &= \mu \mathbf{H} = \mu_r \mu_0 \mathbf{H}
\end{align*}
\]

(2.1.3) (2.1.4)

where \(\varepsilon\) is the electrical permittivity, \(\varepsilon_r\) is the relative permittivity, \(\varepsilon_0\) is the free-space (vacuum) permittivity \((8.854 \times 10^{-12}\) farads/meter\)), \(\mu\) is the magnetic permeability, \(\mu_r\) is the relative permeability, and \(\mu_0\) is the free-space (vacuum) permeability \((4\pi \times 10^{-7}\) henrys/meter\)).
We also have Gauss’ Law for both the electric and magnetic fields, respectively, in an isotropic charge-free medium, given by

\[ \nabla \cdot \vec{D} = 0 \quad (2.1.5) \]
\[ \nabla \cdot \vec{B} = 0. \quad (2.1.6) \]

For simple wave propagation problems in free space, oftentimes the permittivity and permeability are chosen to be constants, which are oftentimes re-scaled to be unity. In metamaterials, however, the permittivity and permeability can be both frequency and spatially dependent, which makes the problem much more difficult to simulate than when they are constant. Since metamaterials are often composite materials, when we refer to the permittivity and permeability, we are actually referring to the effective (relative) permittivity and permeability. Although metamaterial is heterogeneous, it is seen as a single homogenized material with single effective (relative) permittivity and permeability, which may be tensors. Metamaterial is dispersive, i.e. the permittivity and permeability depend on the incoming wave frequency. The lossy Drude model is used by many authors [80, 75] et. al., to describe the material properties of metamaterial. Some use the Lorentz model. These models describe the transport of electrons within the medium. Since metamaterial is inherently a lossy material, it appears that the electrons are not bonded to the atoms, and are free to move around, colliding with one another, as in a conducting material. The Drude model represents this situation. Another model that we did not study in this dissertation, but which is used by some scientists to describe the properties of metamaterial is the Lorentz model. This model presumes that the electrons in a material are more
closely bonded to the atoms, such as in a dielectric material. In this dissertation, we incorporate the lossy Drude model.

Using the lossy Drude polarization and magnetization models in the frequency domain, the electrical permittivity and magnetic permeability, respectively, are described by

\[
\epsilon(\omega) = \epsilon_0 \left(1 - \frac{\omega_{pe}^2}{\omega (\omega - j\Gamma_e)}\right) \\
\mu(\omega) = \mu_0 \left(1 - \frac{\omega_{pm}^2}{\omega (\omega - j\Gamma_m)}\right),
\]

(2.1.7) (2.1.8)

where \(\omega_{pe}\) and \(\omega_{pm}\) are the electric and magnetic plasma frequencies, respectively, \(\Gamma_e\) and \(\Gamma_m\) are the electric and magnetic damping (collision) frequencies, respectively, \(j = -i\), where \(i = \sqrt{-1}\) is the imaginary unit, and \(\omega\) is the general incoming wave frequency.

For the DNG metamaterial case, where permittivity and permeability are only frequency dependent, we use the following constitutive equations:

\[
\vec{D} = \epsilon_0 \vec{E} + \vec{P} = \epsilon_0 \epsilon_r \vec{E} \\
\vec{B} = \mu_0 \vec{H} + \vec{M} = \mu_0 \mu_r \vec{H},
\]

(2.1.9) (2.1.10)

where \(\vec{P}\) is the electric polarization and \(\vec{H}\) is the magnetization. Using a time-harmonic variation of \(exp(j\omega t)\), then from equations 2.1.7 - 2.1.10 we can obtain the corresponding time domain equations for the polarization \(\vec{P}\) and the magnetization \(\vec{M}\) as follows:

\[
\frac{\partial^2 \vec{P}}{\partial t^2} + \Gamma_e \frac{\partial \vec{P}}{\partial t} = \epsilon_0 \omega_{pe}^2 \vec{E} \\
\frac{\partial^2 \vec{M}}{\partial t^2} + \Gamma_m \frac{\partial \vec{M}}{\partial t} = \mu_0 \omega_{pm}^2 \vec{H}.
\]

(2.1.11) (2.1.12)
Noting that the induced electric and magnetic currents, respectively, are

\[ \vec{J} = \frac{\partial \vec{P}}{\partial t}, \]  
\[ \vec{K} = \frac{\partial \vec{M}}{\partial t}, \]  

we obtain the following governing equations for modeling wave propagation in NIMs:

\[ \epsilon_0 \frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H} - \vec{J}, \]  
\[ \mu_0 \frac{\partial \vec{H}}{\partial t} = -\nabla \times \vec{E} - \vec{K}, \]  
\[ \frac{1}{\epsilon_0 \omega^2_{pe}} \frac{\partial \vec{J}}{\partial t} + \frac{\Gamma_e}{\epsilon_0 \omega^2_{pe}} \vec{J} = \vec{E}, \]  
\[ \frac{1}{\mu_0 \omega^2_{pm}} \frac{\partial \vec{K}}{\partial t} + \frac{\Gamma_m}{\mu_0 \omega^2_{pm}} \vec{K} = \vec{H}. \]

For simplicity, we shall assume that the boundary of the domain, \( \Omega \), is perfect conducting, so that

\[ \hat{n} \times \vec{E} = 0 \text{ on } \partial \Omega, \]  

where \( \hat{n} \) is the unit outward normal to \( \partial \Omega \). Furthermore, we assume that the initial conditions are

\[ \vec{E}(\vec{x},0) = \vec{E}_0(\vec{x}), \quad \vec{H}(\vec{x},0) = \vec{H}_0(\vec{x}), \quad \vec{J}(\vec{x},0) = \vec{J}_0(\vec{x}), \quad \vec{K}(\vec{x},0) = \vec{K}_0(\vec{x}). \]  

### 2.2 Basic Mathematical Properties of the Model

In this section, we review some basic properties for our metamaterial model, which were given in [34].
Lemma 2.1. [34, Lemma 2.1] There exists a unique solution for the system (2.1.15)-(2.1.18). Furthermore, the solution of the system (2.1.15)-(2.1.18) satisfies the following stability estimate

\[\epsilon_0 ||\vec{E}(t)||_0^2 + \mu_0 ||\vec{H}(t)||_0^2 + \frac{1}{\mu_0 \omega_{pm}^2} ||\vec{K}(t)||_0^2 + \frac{1}{\epsilon_0 \omega_{pe}^2} ||\vec{I}(t)||_0^2 \leq \epsilon_0 ||\vec{E}(0)||_0^2 + \mu_0 ||\vec{H}(0)||_0^2 + \frac{1}{\mu_0 \omega_{pm}^2} ||\vec{K}(0)||_0^2 + \frac{1}{\epsilon_0 \omega_{pe}^2} ||\vec{I}(0)||_0^2.\] (2.2.1)

Proof. Solving (2.1.17) and (2.1.18) with initial electric and magnetic currents \(\vec{J}_0(\vec{x})\) and \(\vec{K}_0(\vec{x})\) respectively, we obtain

\[\vec{J}(\vec{x}, t; \vec{E}) = e^{-\Gamma_e t} \vec{J}_0(\vec{x}) + \epsilon_0 \omega_{pe}^2 \int_0^t e^{-\Gamma_e (t-s)} \vec{E}(\vec{x}, s) ds \equiv e^{-\Gamma_e t} \vec{J}_0(\vec{x}) + \tilde{\vec{J}}(\vec{E}),\] (2.2.3)

\[\vec{K}(\vec{x}, t; \vec{H}) = e^{-\Gamma_m t} \vec{K}_0(\vec{x}) + \mu_0 \omega_{pm}^2 \int_0^t e^{-\Gamma_m (t-s)} \vec{H}(\vec{x}, s) ds \equiv e^{-\Gamma_m t} \vec{K}_0(\vec{x}) + \tilde{\vec{K}}(\vec{H}),\] (2.2.4)

using which we can rewrite the system (2.1.15)-(2.1.18) as: Find \((\vec{E}, \vec{H})\) such that

\[\epsilon_0 \frac{\partial \vec{E}}{\partial t} - \nabla \times \vec{H} + \tilde{\vec{J}}(\vec{E}) = -e^{-\Gamma_e t} \vec{J}_0(\vec{x}) \quad \forall \ (\vec{x}, t) \in \Omega \times (0, T),\] (2.2.5)

\[\mu_0 \frac{\partial \vec{H}}{\partial t} + \nabla \times \vec{E} + \tilde{\vec{K}}(\vec{H}) = -e^{-\Gamma_m t} \vec{K}_0(\vec{x}) \quad \forall \ (\vec{x}, t) \in \Omega \times (0, T).\] (2.2.6)

To see the existence of the problem (2.2.5)-(2.2.6), we can rewrite it further as

\[\frac{du}{dt} + Au + B \ast u = f, \quad u(0) = u_0,\] (2.2.7)

where we denote the vector \(u = (\epsilon_0 \vec{E}, \mu_0 \vec{H})',\) the vector

\[f = -(\epsilon^{-1}_0 e^{-\Gamma_e t} \vec{J}_0, \mu^{-1}_0 e^{-\Gamma_m t} \vec{K}_0)',\] the linear operator \(A = \left(\begin{array}{cc} 0 & -\mu^{-1}_0 \nabla \times \\ \epsilon^{-1}_0 \nabla \times & 0 \end{array}\right),\)

the memory kernel
\[ B(s) = \begin{pmatrix} \omega_{pe}^2 e^{-\Gamma e s} & 0 \\ 0 & \omega_{pm}^2 e^{-\Gamma m s} \end{pmatrix}, \] and \( B \ast u \) for the convolution between \( B \) and \( u \).

Note that the problem (2.2.7) is a Volterra type equation with the operator \( A \) being an infinitesimal generator of a \( C_0 \)-semigroup on \((L^2(\Omega))^6\), the existence of the solution for (2.2.5)-(2.2.6) is guaranteed \[50\].

The stability can be proved easily. Multiplying equations (2.1.15)-(2.1.18) by \( \vec{E}, \vec{H}, \vec{J}, \vec{K} \) and integrating over the domain \( \Omega \), respectively, then adding the resultants together, we obtain

\[
\frac{1}{2} \frac{d}{dt} \left[ \epsilon_0 ||\vec{E}(t)||_0^2 + \mu_0 ||\vec{H}(t)||_0^2 + \frac{1}{\mu_0 \omega_{pm}^2} ||\vec{K}(t)||_0^2 + \frac{1}{\epsilon_0 \omega_{pe}^2} ||\vec{J}(t)||_0^2 \right] + \frac{\Gamma_m}{\mu_0 \omega_{pm}^2} ||\vec{K}(t)||_0^2 + \frac{\Gamma_e}{\epsilon_0 \omega_{pe}^2} ||\vec{J}(t)||_0^2 = 0,
\]

which directly leads to the stability estimate (2.2.2).

Furthermore, we can prove that the electric and magnetic fields still satisfy Gauss’ law if the initial fields are divergence free. More specifically,

**Lemma 2.2.** [34, Lemma 2.2] Assume that the initial conditions are divergence free, i.e.,

\[
\nabla \cdot (\epsilon_0 \vec{E}_0) = 0, \quad \nabla \cdot (\mu_0 \vec{H}_0) = 0, \quad \nabla \cdot \vec{J}_0 = 0, \quad \nabla \cdot \vec{K}_0 = 0.
\]

Then for any time \( t > 0 \), the electric field \( \vec{E} \) and the electric current \( \vec{J} \) are divergence free. Similarly, for any time \( t > 0 \), the magnetic field \( \vec{H} \) and the magnetic current \( \vec{K} \) are divergence free.

**Proof.** Taking the divergence of (2.1.15), we have

\[
\frac{\partial}{\partial t} \left( \nabla \cdot \epsilon_0 \vec{E} \right) = -\nabla \cdot \vec{J}.
\]
Taking the divergence of (2.1.17), we obtain
\[
\frac{\partial}{\partial t} (\nabla \cdot \vec{J}) + \Gamma_e \nabla \cdot \vec{J} = \omega_{pe}^2 \nabla \cdot \left( \epsilon_0 \vec{E} \right).
\] (2.2.9)

Putting (2.2.8) and (2.2.9) together, we have
\[
\frac{\partial^2}{\partial t^2} (\nabla \cdot \vec{J}) + \Gamma_e \frac{\partial}{\partial t} (\nabla \cdot \vec{J}) + \omega_{pe}^2 \nabla \cdot \vec{J} = 0
\] (2.2.10)

which has the solution \(\nabla \cdot \vec{J}(\vec{x}, t) = 0\) if the initial condition \(\nabla \cdot \vec{J}(\vec{x}, 0) = 0\).

Letting \(\nabla \cdot \vec{J} = 0\) in (2.2.9) leads to \(\nabla \cdot \left( \epsilon_0 \vec{E} \right) = 0\).

Similarly, we can prove \(\nabla \cdot \left( \mu_0 \vec{H} \right) = 0\) and \(\nabla \cdot \vec{K} = 0\).

2.3 Introduction to FDTD Method

The finite-difference time-domain (FDTD) method, originally introduced by Yee in 1966 [58], is still very popular in scientific computing and modeling of electromagnetic fields even today, forty-five years after its invention. It has proven to be robust, and is at the core of many FDTD software constructs. Several companies currently employ its use and continue to create software packages using the FDTD method. Some of these companies are Lumerical Solutions, Inc., which uses FDTD to create microscale optical devices; Acceleware, who provides high performance computer software for oil, gas, and computer engineering markets; Optiwave, who provides FDTD software packages for design and simulation of advanced passive and nonlinear photonic components; and many more. Although there may exist other, more mathematically elegant or even computationally efficient methods, the FDTD method remains popular due to it’s simplistic coding and flexibility.
Yee’s algorithm, based on the time-dependent Maxwell’s curl equations, couples the equations in order to solve for multiple field or field components simultaneously rather than individually. The spatially staggered grid simplifies the contours involved in the curl equations, because it maintains the continuity of the tangential components of the electric and magnetic fields, as well as simplifying the implementation of the boundary conditions. He also used a fully explicit leapfrog scheme in time that also involved second-order central differences, so he staggered the field components temporally. This means the electric field is calculated before or after, but not simultaneously with, the magnetic field. That is the definition of a leapfrog scheme. Using staggered grids both temporally and spatially still provides the most efficient and accurate results when using finite differences.

You can see the spatial staggering of the field components in Figure [2.1] which shows an example of the Yee lattice. The Yee lattice along with central-difference operations implicitly enforce equations (2.1.5) and (2.1.6); thus, it is divergence-free in nature.

The time domain is used instead of the frequency domain in order to allow more flexibility in the frequencies we are able to simulate. Explicit schemes are used to avoid any computationally burdensome matrix inversions, which allows us to solve problems on much larger domains for longer time intervals.

Here, we have incorporated the Yee algorithm, including the staggered grid, using the notation of Taflove [58], and have modified the code of Wagness [58] in many simulations.

The component form of a three-dimensional vector, \( \vec{F} \), is \( \vec{F} = [F_x, F_y, F_z] \). When
we transform the governing equations into component form, equation (2.1.15) becomes

\[
\varepsilon_0 \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - J_x
\]  
(2.3.1)

\[
\varepsilon_0 \frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - J_y
\]  
(2.3.2)

\[
\varepsilon_0 \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - J_z,
\]  
(2.3.3)

equation (2.1.16) becomes

\[
\mu_0 \frac{\partial H_x}{\partial t} = \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - K_x
\]  
(2.3.4)

\[
\mu_0 \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - K_y
\]  
(2.3.5)

\[
\mu_0 \frac{\partial H_z}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - K_z,
\]  
(2.3.6)
equation (2.1.17) becomes

\[
\frac{1}{\epsilon_0 \omega_{pe}^2} \frac{\partial J_x}{\partial t} + \frac{\Gamma_e}{\epsilon_0 \omega_{pe}^2} J_x = E_x \tag{2.3.7}
\]

\[
\frac{1}{\epsilon_0 \omega_{pe}^2} \frac{\partial J_y}{\partial t} + \frac{\Gamma_e}{\epsilon_0 \omega_{pe}^2} J_y = E_y \tag{2.3.8}
\]

\[
\frac{1}{\epsilon_0 \omega_{pe}^2} \frac{\partial J_z}{\partial t} + \frac{\Gamma_e}{\epsilon_0 \omega_{pe}^2} J_z = E_z, \tag{2.3.9}
\]

and equation (2.1.18) becomes

\[
\frac{1}{\mu_0 \omega_{pm}^2} \frac{\partial K_x}{\partial t} + \frac{\Gamma_m}{\mu_0 \omega_{pm}^2} K_x = H_x \tag{2.3.10}
\]

\[
\frac{1}{\mu_0 \omega_{pm}^2} \frac{\partial K_y}{\partial t} + \frac{\Gamma_m}{\mu_0 \omega_{pm}^2} K_y = H_y \tag{2.3.11}
\]

\[
\frac{1}{\mu_0 \omega_{pm}^2} \frac{\partial K_z}{\partial t} + \frac{\Gamma_m}{\mu_0 \omega_{pm}^2} K_z = H_z. \tag{2.3.12}
\]

The two-dimensional \( TE_z \) mode yields only the components \( E_x, E_y, \) and \( H_z \) by assuming the structure being modeled extends to infinity in the \( z \)-direction, equivalent to setting all partial derivatives with respect to \( z \) equal to zero. Then equations (2.3.1)-(2.3.12) reduce to

\[
\epsilon_0 \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y} - J_x \tag{2.3.13}
\]

\[
\epsilon_0 \frac{\partial E_y}{\partial t} = -\frac{\partial H_z}{\partial x} - J_y \tag{2.3.14}
\]

\[
\mu_0 \frac{\partial H_z}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - K_z \tag{2.3.15}
\]

\[
\frac{1}{\epsilon_0 \omega_{pe}^2} \frac{\partial J_x}{\partial t} + \frac{\Gamma_e}{\epsilon_0 \omega_{pe}^2} J_x = E_x \tag{2.3.16}
\]

\[
\frac{1}{\epsilon_0 \omega_{pe}^2} \frac{\partial J_y}{\partial t} + \frac{\Gamma_e}{\epsilon_0 \omega_{pe}^2} J_y = E_y \tag{2.3.17}
\]

\[
\frac{1}{\mu_0 \omega_{pm}^2} \frac{\partial K_z}{\partial t} + \frac{\Gamma_m}{\mu_0 \omega_{pm}^2} K_z = H_z \tag{2.3.18}
\]
2.4 The Yee Algorithm

Using finite differences has been traced back as early as 1910, when it was used for weather forecasting. In the 1920s, it was referred to as the "method of squares" by A. Thom [73]. We can discretize differential equations by use of Taylor series expansion about the point we want to approximate the derivative at (whether it be in time or space) by writing the derivative at that location in terms of the values of its neighboring points. In order to do this, we need a grid of points called nodes.

If we let $F$ now represent any electromagnetic field component, then we will use the notation as in [58] that $F(i \Delta x, j \Delta y, k \Delta z, n \Delta t) = F_{i,j,k}^n$, where $\Delta x$, $\Delta y$, and $\Delta z$ are the mesh sizes in the $x$, $y$, or $z$-direction, respectively, and $\Delta t$ is the time increment. So, the second order finite difference for the first partial derivative of $F$ in the $x$-direction evaluated at time $t_n = n \Delta t$ is

$$\frac{\partial F}{\partial x}(i \Delta x, j \Delta y, k \Delta z, n \Delta t) = \frac{F_{i+\frac{1}{2},j,k}^n - F_{i-\frac{1}{2},j,k}^n}{\Delta x} + O[\Delta x^2].$$

The same pattern is true for all other partial derivatives.

2.5 Numerical Dispersion and Stability

The Yee algorithm is conditionally stable, and the maximum allowable time step $\Delta t_{\text{max}}$ in order for the algorithm to remain stable is limited by

$$\Delta t_{\text{max}} \leq \frac{1}{c \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}},$$

where $c$ is the speed of light, which is the maximum velocity any wave can travel within a vacuum. In one dimension, the Courant number is defined as

$$S \equiv \frac{c \Delta t}{\Delta x}.$$
Then, we see that if equation (2.5.1) is written as

\[ c\Delta t_{\text{max}} \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}} \leq 1, \quad (2.5.2) \]

and if \( \Delta x = \Delta y = \Delta z \), then \( S = 1 \) in the 1-D case, \( S = \frac{1}{\sqrt{2}} \) for the 2-D case, and \( S = \frac{1}{\sqrt{3}} \) for the 3-D case to remain stable.

### 2.6 Absorbing Boundary Conditions

Simulating wave propagation from scattering, antennas, or waveguides requires an unbounded domain or a domain large enough so that waves do not reflect off the domain boundaries back into the computational domain and interfere with the wave propagation being analyzed. The computational requirements for making a domain large enough to prevent these reflections would be nearly computationally impossible, or at least highly undesirable, in most cases. It is impossible to have an unbounded domain in scientific computing. Therefore, scientists have come up with absorbing boundary conditions (ABCs). Normally, this is an additional domain surrounding in part or wholly the computational domain. It is connected to the computational domain boundary, but the fields are computed separately in this external domain. All tangential properties are preserved, and the fields are continuous across the boundary. It is made so that it absorbs waves that come in contact with that region. There are several different ABCs, but one of the most popular is that proposed by J.P. Berenger, which is called the perfectly matched layer (PML). It is designed to treat the external region as a lossy material. So, it doesn’t really absorb the wave as it does severely dampen the wave (causes the wave to attenuate) as it enters that region by removing
its power so quickly that there’s nothing left to reflect back off the outer boundary.

The original Berenger’s PML for the two-dimensional $TE_z$ mode case splits $H_z$ into two subcomponents, so that $H_z = H_{zx} + H_{zy}$. More information can be found in ??.

### 2.7 Dispersive FDTD Scheme for Metamaterial Simulation

In keeping with the same notation used in [58, p. 92], equations (2.3.13)-(2.3.18) can now be written in their discretized forms.

\[
\begin{align*}
\varepsilon_0 \frac{\partial E_x}{\partial t} |_{i,j+\frac{1}{2}}^{n} & = \frac{\partial H_z}{\partial y} |_{i,j+\frac{1}{2}}^{n} - J_x |_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \quad (2.7.1) \\
\varepsilon_0 \frac{\partial E_y}{\partial t} |_{i-\frac{1}{2},j+1}^{n} & = - \frac{\partial H_z}{\partial x} |_{i-\frac{1}{2},j+1}^{n} - J_y |_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} \quad (2.7.2) \\
\mu_0 \frac{\partial H_z}{\partial t} |_{i,j+1}^{n+\frac{1}{2}} & = \frac{\partial E_x}{\partial x} |_{i,j+1}^{n+\frac{1}{2}} - \frac{\partial E_y}{\partial x} |_{i,j+1}^{n+\frac{1}{2}} - K_z |_{i,j+1}^{n+\frac{1}{2}} \quad (2.7.3) \\
\frac{1}{\varepsilon_0 \omega_{pe}^2} \frac{\partial J_x}{\partial t} |_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Gamma_e}{\varepsilon_0 \omega_{pe}^2} J_x |_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} & = E_x |_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \quad (2.7.4) \\
\frac{1}{\varepsilon_0 \omega_{pe}^2} \frac{\partial J_y}{\partial t} |_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} + \frac{\Gamma_e}{\varepsilon_0 \omega_{pe}^2} J_y |_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} & = E_y |_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} \quad (2.7.5) \\
\frac{1}{\mu_0 \omega_{pm}^2} \frac{\partial K_z}{\partial t} |_{i,j+1}^{n+1} + \frac{\Gamma_m}{\mu_0 \omega_{pm}^2} K_z |_{i,j+1}^{n+1} & = H_z |_{i,j+1}^{n+1} \quad (2.7.6)
\end{align*}
\]

Since the scheme for the first three equations is already given in [58], we focus on the remaining three equations (2.7.4)-(2.7.6). Using the second-order Taylor expansion in time on the first term and noting that the time locations must be consistent, we use averaging in time on the second term, which gives

\[
\frac{J_x |_{i,j+\frac{1}{2}}^{n+1} - J_x |_{i,j+\frac{1}{2}}^{n}}{\Delta t} + \frac{\Gamma_e}{2} \left( J_x |_{i,j+\frac{1}{2}}^{n+1} + J_x |_{i,j+\frac{1}{2}}^{n} \right) = \varepsilon_0 \omega_{pe}^2 E_x |_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}. \quad (2.7.7)
\]
Simplification results in
\[
\left(1 + \frac{1}{2} \Gamma_e \Delta t\right) J_x|_{i,j+\frac{1}{2}}^{n+1} + \left(-1 + \frac{1}{2} \Gamma_e \Delta t\right) J_x|_{i,j+\frac{1}{2}}^n = \Delta t \varepsilon_0 \omega_p^2 E_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}. \tag{2.7.8}
\]

By similar algebra, we obtain the following updating equation for \( J_y \) from equation (2.7.5) and \( K_z \) from equation (2.7.6), respectively:
\[
\left(1 + \frac{1}{2} \Delta t \Gamma_e\right) J_y|_{i-\frac{1}{2},j+1}^{n+1} + \left(-1 + \frac{1}{2} \Delta t \Gamma_e\right) J_y|_{i-\frac{1}{2},j+1}^n = \Delta t \varepsilon_0 \omega_p^2 E_y|_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} \tag{2.7.9}
\]
\[
\left(1 + \frac{1}{2} \Delta t \Gamma_m\right) K_z|_{i,j+1}^{n+1} + \left(-1 + \frac{1}{2} \Delta t \Gamma_m\right) K_z|_{i,j+1}^n = \Delta t \mu_0 \omega_{pm}^2 H_z|_{i,j+1}^{n+\frac{1}{2}} \tag{2.7.10}
\]

Inside the metamaterial, we use the following numerical scheme:
\[
E_{x,i,j+\frac{1}{2}}^{n+\frac{1}{2}} = C_{a,E_x} E_{x,i,j+\frac{1}{2}}^n + C_{b,E_x} E_{x,i,j+\frac{1}{2}}^n \left(H_{z,i,j+\frac{1}{2}}^n - H_{z,i,j}^n\right) - \cdots \tag{2.7.11}
\]
\[
dx \times J_x|_{i,j+\frac{1}{2}}^n
\]
\[
E_{y,i-\frac{1}{2},j+1}^{n+\frac{1}{2}} = C_{a,E_y} E_{y,i-\frac{1}{2},j+1}^n + C_{b,E_y} E_{y,i-\frac{1}{2},j+1}^n \left(H_{z,i-1,j+1}^n - H_{z,i,j+1}^n\right) - \cdots \tag{2.7.12}
\]
\[
dx \times J_y|_{i-\frac{1}{2},j+1}^n
\]
\[
H_{z,i,j+1}^{n+1} = D_{a,H_z} H_{z,i,j+1}^n H_{z,i,j+\frac{1}{2}}^n + \cdots \tag{2.7.13}
\]
\[
D_{b,H_z} E_{x,i,j+\frac{1}{2}}^n - E_{x,i,j+\frac{1}{2}}^n + E_{y,i-\frac{1}{2},j+1}^n - E_{y,i-\frac{1}{2},j+1}^n - \cdots \tag{2.7.14}
\]
\[
dx \times K_z|_{i,j+\frac{1}{2}}^{n+1}
\]
\[
J_x|_{i,j+\frac{1}{2}}^{n+1} = \frac{1 - \frac{1}{2} \Gamma_e \Delta t}{1 + \frac{1}{2} \Gamma_e \Delta t} J_x|_{i,j+\frac{1}{2}}^n + \frac{\Delta t \varepsilon_0 \omega_p^2}{1 + \frac{1}{2} \Gamma_e \Delta t} E_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \tag{2.7.14}
\]
\[
J_y|_{i-\frac{1}{2},j+1}^{n+1} = \frac{1 - \frac{1}{2} \Delta t \Gamma_e}{1 + \frac{1}{2} \Delta t \Gamma_e} J_y|_{i-\frac{1}{2},j+1}^n + \frac{\Delta t \varepsilon_0 \omega_p^2}{1 + \frac{1}{2} \Delta t \Gamma_e} E_y|_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} \tag{2.7.15}
\]
\[
K_z|_{i,j+1}^{n+1} = \frac{1 - \frac{1}{2} \Delta t \Gamma_m}{1 + \frac{1}{2} \Delta t \Gamma_m} K_z|_{i,j+1}^n + \frac{\Delta t \mu_0 \omega_{pm}^2}{1 + \frac{1}{2} \Delta t \Gamma_m} H_z|_{i,j+1}^{n+\frac{1}{2}} \tag{2.7.16}
\]

where the coefficients \( C_{a,E_x}, C_{b,E_x}, C_{a,E_y}, C_{b,E_y}, D_{a,H_z}, \) and \( D_{b,H_z} \) are given in [58, p. 85]. Outside the metamaterial, in freespace, we use the traditional FDTD scheme.
in [58] along with Berenger’s perfectly matched layer (PML) around the entire computational domain.

2.8 Numerical Simulations

In [80], simulations are performed for DNG metamaterial for two cases: when the index of refraction, \( n \approx -1 \) and when \( n \approx -6 \). The general index of refraction, \( n_i \), where the index \( i \) stands for the medium the wave is propagating through, is given by the expression

\[
n_i = \sqrt{\frac{\epsilon_i}{\epsilon_0}} \sqrt{\frac{\mu_i}{\mu_0}} = \sqrt{\epsilon_r \mu_r}
\]

Since both \( \epsilon_r \) and \( \mu_r \) are negative, then \( n_i \) is negative, as well.

In addition, he used the following lossy Drude polarization and magnetization models:

\[
\epsilon(\omega) = \epsilon_0 \left(1 - \frac{\omega_{pe}^2}{\omega(\omega - j\Gamma_e)} \right) \tag{2.8.1}
\]

\[
\mu(\omega) = \mu_0 \left(1 - \frac{\omega_{pm}^2}{\omega(\omega - j\Gamma_m)} \right) \tag{2.8.2}
\]

In Ziolkowski’s simulations with a normally incident wave, he simulated the two-dimensional propagation of an s-Polarized field of a \( TE \) wave, using the field components \( H_x, H_z, \) and \( E_y \). s-Polarized meaning that light is polarized perpendicular to the plane of incidence. In our simulations, we chose a \( TE_z \) wave which consists of the field components \( E_x, E_y, \) and \( H_z \). Wave impedance is given by \( \eta_i = \frac{\sqrt{\mu_i}}{\sqrt{\epsilon_i}} \), where \( i \) denotes the particular media. If we let \( \text{trans} \) mean transmitted and \( \text{inc} \) means incident, a ”matched slab” is defined by \( \eta_{\text{trans}} = \eta_{\text{inc}} \). For simulations of both cases, \( n \approx -1 \) and \( n \approx -6 \), matched slabs are considered, in which case \( \omega_{pe} = \omega_{pm} = \omega_p \) and
\[ \Gamma_e = \Gamma_m = \Gamma. \] In all cases, \( \Gamma = 10^8 \, s^{-1} \), frequency is \( f_0 = 30 \, GHz \), the mesh size, \( \Delta x = \frac{\lambda_0}{100} = 100 \, \mu m \), time step, \( \Delta t = 22.39 \, ps = 22.39 \times 10^{-12} \, s \), and simulation domain size is \( 640 \times 830 \) cells. Also, for both cases, multiple cycle \( m - n - m \) pulses were used to generate the source that are smooth thus producing minimal noise. These pulses are given by the expressions

\[ f(t) = \begin{cases} 
0 & \text{for } t < 0 \\
g_{\text{on}}(t) \sin(\omega t) & \text{for } 0 < t < mT_p \\
\sin(\omega t) & \text{for } mT_p < t < (m + n)T_p \\
g_{\text{off}}(t) \sin(\omega t) & \text{for } (m + n)T_p < t < (m + n + m)T_p \\
0 & \text{for } (m + n + m)T_p < t 
\end{cases} \]  

(2.8.3)

where \( x_{\text{on}}(t) = \frac{t}{mT_p} \), \( x_{\text{off}} = \frac{t - (m + n)T_p}{mT_p} \), and the continuous functions in \( C^2 \) are

\[ g_{\text{on}} = 10x_{\text{on}}^3(t) - 15x_{\text{on}}^4(t) + 6x_{\text{on}}^5(t) \]

\[ g_{\text{off}} = 1 - \left[ 10x_{\text{off}}^3(t) - 15x_{\text{off}}^4(t) + 6x_{\text{off}}^5(t) \right]. \]

For the case when \( n \approx -1 \), \( \omega_p = 2\pi\sqrt{2}f_0 \approx 2.66573 \times 10^{11} \, rad/s \), so \( \Gamma = 3.75 \times 10^{-4}\omega_p \). One can easily see that given this value of \( \omega_p \), that the real part of both \( \epsilon_r \) and \( \mu_r \) is \(-1\), and thus \( n \approx -1 \) since \( n = \sqrt{\epsilon_r\sqrt{\mu_r}} \). Similarly, when \( n \approx -6 \), \( \omega_p = 2\pi\sqrt{7}f_0 \approx 4.98712 \times 10^{11} \, rad/s \), so \( \Gamma = 2.01 \times 10^{-4}\omega_p \). For both cases, we used domain size \( 640 \times 830 \) cells. The Gaussian beam varies spatially as \( \exp(-x^2/w_0^2) \), where \( w_0 \) is the "waist", which is set to 50. The single slab has a depth of \( 2\lambda_0 = 20 \, cells \) and width of \( 6\lambda_0 = 600 \, cells \). The source is placed 40 cells from the bottom and 200 cells away from the slab.
Figure 2.2. Single slab metamaterial with $n \approx -1$. 
Figure 2.3. Single slab metamaterial with $n \approx -6$. 
An extension on Ziolkowski’s paper [80] shows the nice refocusing property of the DNG metamaterial shown in Figure 2.4 using multiple DNG metamaterial slabs. The domain size of this simulation is $500 \times 1500$ cells. Each slab is 200 cells thick and 460 cells wide and are separated by 400 cells. The refocusing property assists a beam in being transmitted to far distances using multiple DNG metamaterial slabs, which has the potential in applications involving nano waveguides.

In another extension, we simulate the scattering of the Gaussian wave from a DNG metamaterial circle, which represents the cross-section of an infinite-length cylinder, which could be a wire.
Figure 2.4. Three slabs of metamaterial with $n \approx -1$ representing a possible waveguide.
Figure 2.5. Cylindrical metamaterial of radius $0.25\lambda_0$ with $n \approx -1$. 
Figure 2.6. Cylindrical metamaterial of radius $0.5\lambda_0$ with $n \approx -1$. 
Figure 2.7. Cylindrical metamaterial of radius $\lambda_0$ with $n \approx -1$. 
Figure 2.8. Cylindrical metamaterial of radius $0.25\lambda_0$ with $n \approx -6$. 
Figure 2.9. Cylindrical metamaterial of radius $0.5\lambda_0$ with $n \approx -6$. 
Figure 2.10. Cylindrical metamaterial of radius $\lambda_0$ with $n \approx -6$. 
CHAPTER 3
SIMULATION OF CLOAKING PHENOMENON

3.1 Introduction to Cloaking

The creation of metamaterials has given rise to the development of cloaking materials: materials which bend electromagnetic waves around objects so that the object appears invisible. Ideally, we want the waves to pass through the object and around the hidden object so that the presence of the hidden object is undetectable, meaning there is no phase change, energy loss, or trajectory change. Many scientists hope this can be done using transformation optics.

One of the pioneers in transformation optics is J.B. Pendry. Transformation optics involves bending light, EM waves, or energy into any direction or shape that suits your needs. Pendry, Schurig, and Smith wrote a paper discussing the hopes that metamaterials can be made so that the material parameters can vary independently and arbitrarily [49].

\[ \text{Figure 3.1. Ray trajectories of the flow of EM waves through a metamaterial cloak. (A) 2-D cross section, (B) 3-D cross section [49]} \]
We want the hidden object to lie inside a sphere of radius \( R_1 \) with the cloaking region to be the annulus \( R_1 < r' < R_2 \). Electromagnetic waves that would normally travel directly through the whole sphere, \( 0 \leq r \leq R_2 \), would then be redirected to travel only through the spherical shell by use of the following linear transformation:

\[
r' = R_1 + r(R_2 - R_1)/R_2 \tag{3.1.1}
\]

\[
\theta' = \theta \tag{3.1.2}
\]

\[
\phi' = \phi. \tag{3.1.3}
\]

By applying basic coordinate transformation techniques and normalization, we arrive at the following transformed material parameters:

\[
\epsilon_{r'} = \mu_{r'} = \frac{R_2}{R_2 - R_1} \left( \frac{r' - R_1}{(r')^2} \right) \tag{3.1.4}
\]

\[
\epsilon_{\theta'} = \mu_{\phi'} = \frac{R_2}{R_2 - R_1} \tag{3.1.5}
\]

\[
\epsilon_{\phi'} = \mu_{\phi'} = \frac{R_2}{R_2 - R_1}. \tag{3.1.6}
\]

I have included the proof of the above transformation in Appendix C for those who are interested. Notice that in the above set of parameters, only \( \epsilon_{r'} \) is radially dependent.

Maxwell’s equations are form invariant, thus they take on the same form after a coordinate transform. Only the material parameters change, and often become tensors in this case.

The method used above in [49] may be impractical, however, because (1) the effect is only achieved at one frequency [49], and (2) full knowledge of permittivity and permeability of the material is required [75].
It is notable that the science that is involved in creating electromagnetic cloaks is very beneficial to improve the technology already being used today, such as radar, cell phones, and wireless Internet.

Since most methods are applied in the frequency domain, such as finite-element method (FEM), Zhao et. al. [75] propose a dispersive FDTD method where the material parameters follow the Drude model. Their method takes frequency and radial-dependence of material parameters into consideration.

Pendry continued to help others propose additional electromagnetic cloaks using coordinate transformations, such as in [51]. They use a form-invariant spacial coordinate transformation which in turn transforms the relative permittivity and permeability tensors in anisotropic media.

In 2006, Dr. Smith and other scientists at Duke realized a metamaterial cloak of invisibility over a narrow frequency band in the microwave range. The cloak “hid” a narrow copper cylinder. Pictures from their results are shown in Figures 3.2 and 3.3.

An interesting creation has been developed in the UK in 2010 called MetaFlex. This is a flexible metamaterial they hope to be able to use someday to create a cloak. An effective tailored response can be obtained by layering MetaFlex. Figures 3.4 and 3.5 show samples of their realizations.

3.2 FDTD Method for Cloaking

The Yee FDTD algorithm only deals with frequency-independent materials, so it is necessary to create a new FDTD scheme to handle frequency-dependent materials. Those discussed in [51] are frequency and radially dependent. There is now
Figure 3.2. 2-D microwave cloaking structure. First realization of a cloak by Smith and his group at Duke in 2006 [53].

Figure 3.3. (A) Simulation results with exact material parameters, (B) Simulation results with reduced parameters, (C) Experimental measurement of the bare cylinder, and (D) Experimental measurement of the cloaked cylinder [53].

an abundance of papers discussing metamaterial cloaks, but the majority of them perform their simulations in the frequency domain. Simulations have been done us-
ing finite element method and via a commercial simulation package called COMSOL Multiphysics™, but those had primarily been done in the frequency domain, as well. There are a few different dispersive FDTD methods, such as the recursive convolution (RC) method, the auxiliary differential equation (ADE) method, and the Z-transform method. The ADE method has been used in [51] for simplicity.

In 2008, a new radially-dependent dispersive FDTD scheme was proposed for modeling electromagnetic cloaking structures [51]. Since the materials used to build
cloaks are metamaterials, which are inherently dispersive, they are therefore band-limited. Other cloak models which have been considered are coordinate transformation techniques, as the one we looked at earlier done in [49], cylindrical wave expansion techniques, and full-wave Mie scattering models to name a few.

We assume an axisymmetric cylindrical cloak, and consider a two-dimensional cross-section. The cloak has inner radius $R_1$ and outer radius, $R_2$, and the hidden object lies within the area $r < R_1$. The cloaking metamaterial is between $R_1$ and $R_2$, and freespace (or other medium) lies in the region $r > R_2$. The material parameters of the anisotropic material are radially dependent and are given by

$$
\epsilon_r = \frac{r - R_1}{r}, \quad \epsilon_\phi = \frac{r}{r - R_1}, \quad \mu_z = \left(\frac{R_2}{R_2 - R_1}\right)^2 \frac{r - R_1}{r}.
$$

(3.2.1)

Using the Drude model to map the material parameters, $\epsilon_r(\omega) = 1 - \frac{\omega_p^2}{\omega^2 - j\omega\gamma}$, where $\omega_p$ is the plasma frequency and $\gamma$ is the collision frequency of the material, and varying the plasma frequency allows the radial dependence of the material parameters.

Since the material parameters are radially-dependent, but we are using FDTD, which requires a rectangular mesh, the following standard cylindrical to Cartesian coordinate transformations are used, as shown here:

$$
\begin{bmatrix}
\epsilon_{xx} & \epsilon_{xy} \\
\epsilon_{yx} & \epsilon_{yy}
\end{bmatrix} = \begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix}
\epsilon_r & 0 \\
0 & \epsilon_\phi
\end{bmatrix} \begin{bmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{bmatrix}
$$

$$
= \begin{bmatrix}
\epsilon_r \cos^2 \theta + \epsilon_\phi \sin^2 \theta & (\epsilon_r - \epsilon_\phi) \sin \theta \cos \theta \\
(\epsilon_r - \epsilon_\phi) \sin \theta \cos \theta & \epsilon_r \sin^2 \theta + \epsilon_\phi \cos^2 \theta
\end{bmatrix}
$$

(3.2.2)

(3.2.3)

This is the permittivity tensor to be used in our simulations.

We begin with the governing equations (2.1.15) and (2.1.16) with $\vec{J} = \vec{K} = 0$ and the constitutive equations (2.1.3) and (2.1.4) to develop the basic scheme.
From Ampere’s Law, we derive the updating equations for $\vec{D}$, the electric flux density

$$D_x^{n+\frac{1}{2}}|_{i,j+\frac{1}{2}} = D_x^{n-\frac{1}{2}}|_{i,j+\frac{1}{2}} + \frac{\Delta t}{\Delta x} \left( H_z^n|_{i,j+1} - H_z^n|_{i,j} \right)$$ \hspace{1cm} (3.2.4)

$$D_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} = D_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} - \frac{\Delta t}{\Delta x} \left( H_z^n|_{i,j+1} - H_z^n|_{i-1,j+1} \right) .$$ \hspace{1cm} (3.2.5)

From Faraday’s Law, we derive the updating equation for $\vec{B}$, the magnetic flux density

$$B_z^{n+1}|_{i,j+1} = B_z^n|_{i,j+1} + \frac{\Delta t}{\Delta x} \left( E_x^{n+\frac{1}{2}}|_{i,j+\frac{1}{2}} - E_x^{n+\frac{1}{2}}|_{i,j+\frac{1}{2}} + E_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} - E_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} \right) .$$ \hspace{1cm} (3.2.6)

From the constitutive equations (2.1.3) and (2.1.4), we derive the following updating equations for the electric and magnetic fields:

$$E_x^{n+\frac{1}{2}}|_{i,j+\frac{1}{2}} = \frac{1}{l_x} \left[ a_x D_x^{n+\frac{1}{2}} + b_x D_x^{n-\frac{1}{2}} + c_x D_x^{n-\frac{3}{2}} + \cdots \right] \left( g_x E_x^{n-\frac{1}{2}} + h_x E_x^{n-\frac{3}{2}} \right) i,j+\frac{1}{2} + \cdots$$ \hspace{1cm} (3.2.7)

$$\vec{D}_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} = \frac{1}{4} \left\{ D_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j} + D_y^{n+\frac{1}{2}}|_{i+\frac{1}{2},j} + D_y^{n+\frac{1}{2}}|_{i+\frac{1}{2},j+1} + D_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} \right\}$$ \hspace{1cm} (3.2.8)

$$E_y^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} = \frac{1}{l_y} \left[ a_y D_y^{n+\frac{1}{2}} + b_y D_y^{n-\frac{1}{2}} + c_y D_y^{n-\frac{3}{2}} + \cdots \right] \left( g_y E_y^{n-\frac{1}{2}} + h_y E_y^{n-\frac{3}{2}} \right) i-\frac{1}{2},j+1 + \cdots$$ \hspace{1cm} (3.2.9)

$$\vec{D}_x^{n+\frac{1}{2}}|_{i-\frac{1}{2},j+1} = \frac{1}{4} \left\{ D_x^{n+\frac{1}{2}}|_{i,j+\frac{1}{2}} + D_x^{n+\frac{1}{2}}|_{i,j+\frac{1}{2}} + D_x^{n+\frac{1}{2}}|_{i-1,j+\frac{1}{2}} + D_x^{n+\frac{1}{2}}|_{i-1,j+\frac{1}{2}} \right\} .$$ \hspace{1cm} (3.2.10)

where all the coefficients are defined in [51], p. 6723. The calculation of $\vec{D}_x$ and $\vec{D}_y$ are derived in [29] originally for what appears to be a non-staggered grid. We derived the above formulas based on the original derivation.
3.3 Numerical Simulations

The 2-D FDTD simulation with cell size $\Delta x = \Delta y = \lambda/150$, where $\lambda$ can be derived by using the operating frequency, $f = 2.0\text{GHz}$. The time step meets the usual Courant stability requirement, so $\Delta t = \Delta x/\sqrt{2c}$, where $c$ is the speed of light. Assuming the ideal lossless case, $\gamma = 0$. The internal radius of the cloak is $R_1 = 0.1m$ and the external radius of the cloak is $R_2 = 0.2m$. A sinusoidal "hard" line source is used, where a whole column of the field $H_z$ is updated with the source continually in order to propagate the wave through the cloaking region. The Berenger PML is used to truncate the domain.

For all of the examples below, it took approximately 370s for only the parameters and coefficients to be updated. See Appendix B.

In all of the following examples, an ideal cloak is considered, with the material parameters $\mu_r = \mu_\phi = \epsilon_z = 0$.

3.4 Example 1

In the case where $\mu_r = \mu_\phi = \epsilon_z = 0$, with these parameters defined in equation (3.2.1), we show the results for the magnetic field, $H_z$ only, in Figure 3.6. Note that all material parameters here are radially dependent.

The running time for the main time loop without any graphics display was approximately 38,473s, or nearly 11 hours on a domain of size $800 \times 800$. 

45
3.5 Example 2

Since all material parameters in Example 1 are radially dependent, it is unrealistic, as creating an ideal cloak whose parameters all depend on radius is unrealizable. In that case, this example has a reduced set of material parameters. Here, they are given by

$$
\epsilon_r = \left( \frac{R_2}{R_2 - R_1} \right)^2 \left( \frac{r - R_1}{r} \right)^2, \quad \epsilon_\phi = \left( \frac{R_2}{R_2 - R_1} \right)^2, \quad \mu_z = 1 \quad (3.5.1)
$$

We ran the simulation on a domain of size 800 × 800 cells with enough time steps...
to see the fields reach a steady-state.

We can see from Figures 3.7 and 3.8 that the $E_x$ field is very good, whereas the $H_z$ field has some ripples, which indicate reflections. This is because the impedance matching at the outer boundary of the cloak is no longer satisfied.

Figure 3.7. Example 2: $E_x$ field of the cylindrical cloak with reduced material parameters.
3.6 Example 3

Due to reflection in the previous example (the simplified cloak), we now use the high-order transformation given in [9], which will reduce the scattering seen in Example 2. Here, the material parameters are given by

\[
\epsilon_r = \left( \frac{r'}{r} \right)^2, \quad \epsilon_\phi = \left[ \frac{\partial g(r')}{\partial r'} \right]^{-2}, \quad \mu_z = 1,
\]

(3.6.1)

where

\[
r = g(r') = [(R_1/R_2)(r'/R_2 - 2) + 1]r' + R_1
\]

(3.6.2)
In [9], the requirement is that $R_1/R_2 < 0.5$, so the dimensions of the cloak are at its limit since $R_2 = 2R_1$.

We wrote the simulation in terms of $r'$, where $0 \leq r' \leq R_2$. These simulations are shown here. One must transform these values back to $r$, which lies in $R_1 \leq r \leq R_2$, by using the inverse transform of (3.6.2).

These simulation figures are with respect to $r'$ and must be transformed back to $r$ to show the cloak in the proper region.
Figure 3.9. Example 3: $E_x$ field of an improved cylindrical cloak using a high-order transformation.
Figure 3.10. Example 3: $H_z$ field of an improved cylindrical cloak using a high-order transformation.
CHAPTER 4
HIGH ORDER COMPACT SCHEMES FOR MAXWELL’S EQUATIONS

4.1 Introduction to Compact Difference Schemes

Compact difference schemes have been introduced and used in order to reduce the accumulation of errors in simulations over longer period of times and farther distances. This is useful in simulations of many applications such as aeroacoustics and electromagnetics. One of the earliest pioneers in compact finite difference schemes was Lele [30]. His paper, written in 1992, focused not only on the truncation error but mainly on using finite difference schemes which have spectral-like resolution, but which were more easily adaptable to more complex domains and boundary conditions as an alternative to spectral methods. The schemes begin as generalizations of Pade’ schemes. More recently, in 2006, Li and Visbal coupled the compact difference schemes developed by Lele with a high-order low-pass filter to simulate nonlinear dispersive waves [36].

It is well known, and has been noted in [22, 60] et. al., that staggered grid schemes provide increased accuracy when solving wave equations, such as Maxwell’s Equations, which are often posed as a system of ordinary differential equations (ODEs). It is possible to have staggering in both time and space, as we have seen in the basic second-order Yee scheme [58]. Several people have studied high order schemes in both space and time. In particular, in [22], Ghrist et. al. introduce a fourth-order multi-stage method which is comparable to the classic fourth-order Runge-Kutta method, denoted RK4. Their new method is 16 times more accurate than RK4 and allows for time steps about twice as large as those of RK4 with an equivalent amount of
4.2 High Order Spatial Derivatives for Staggered Grids

If we consider the governing equations (2.1.15) and (2.1.16), we see from the curl that only first partial derivatives are necessary to calculate in order to solve the system. In the original Yee algorithm, a second-order central difference method was employed to compute the first partial derivatives in any direction. We now consider higher order, in particular fourth order, schemes, which we ultimately want to use to solve Maxwell’s equations in complex media. We are going to consider only those that are defined and useful on a spatially staggered grid.

Lele offers a scheme to calculate the derivatives at cell centers, which are a half-cell \((h/2)\) from the nodes on which the function values are given. For example, if the function values are provided at the \(x_i\) nodes, where \(i\) are integer values, then \(h = x_i - x_{i-1}\), and the cell centers are located at \(x_{i \pm 1/2}\). The diagram in Figure 4.1 shows node locations for a 1-D staggered grid. These formulations are highly desirable for use on a staggered grid, which is primarily used in simulations of electromagnetic wave propagation. Lele points out that the cell-centered grids have better resolution characteristics for wavenumbers near \(\pi\) than the non-staggered grids he derives in his paper [30].

An approximation of the first derivative at a cell center, with \(h\) being the cell size, is given by

\[
\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = \frac{c(f_{i+\frac{1}{2}} - f_{i-\frac{3}{2}})}{5h} + \frac{b(f_{i+\frac{3}{2}} - f_{i-\frac{1}{2}})}{3h} + \frac{a(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}})}{h}.
\]

(4.2.1)
Here, the subscript $i$ denotes the location of the node that we wish to differentiate. We can solve for the coefficients, $a$, $b$, and $c$ by matching the Taylor series coefficients. If $\beta = c = 0$, tridiagonal schemes analogous to the standard Padé scheme are obtained. Then, the remaining coefficients would be

$$\beta = 0 \quad a = \frac{3}{8}(3 - 2\alpha) \quad b = \frac{1}{8}(22\alpha - 1).$$

(4.2.2)

It can easily be shown that the truncation error for the right hand side of Equation (4.2.1) is $((9 - 62\alpha)/1920)h^4 f^{(5)}$.

In 1997, Turkel and Yefet published a fourth-order compact scheme for Maxwell’s Equations on a staggered mesh [60]. Their methods are named $\text{Ty}(t,s)$, where $t$ stands for the temporal order and $s$ the spatial order of the scheme. After publishing this paper, Eli Turkel went on to contribute chapters in the follow-up book to Taflove’s first book [58]. In the follow-up book [59], Turkel included information about his high-order compact schemes.

A popular explicit fourth-order staggered scheme is given by

$$\frac{\partial u_i}{\partial x_j} = \frac{27}{24\Delta x_j} \left( \frac{u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}}{\Delta x_j} - \frac{u_{i+\frac{3}{2}} - u_{i-\frac{3}{2}}}{\Delta x_j} \right),$$

(4.2.3)

where $i$ stands for the node location you want to differentiate at, and $x_j = \{x, y, z\}$. 

54
Note here that you must have two nodes available on each side of node \( i \) in order to calculate the derivative at node \( i \). This means that for some field components in the discretized component form of Maxwell’s Equations, it will not be possible to calculate the derivative at the boundary nodes, nor at some of the near-boundary nodes using this formula.

Turkel also developed an implicit fourth-order staggered scheme [60], which is given by

\[
\frac{u'_{i+1} + u'_{i-1}}{24} + \frac{11}{12}u'_i = \frac{u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}}{\Delta x_j},
\]

where \( x_j = \{x, y, z\} \). Since the implicit implementation includes one neighboring node on each side of node \( i \), this scheme can also only be used for internal grid points.

In order to solve for the derivative at the first node, \( u_{\frac{3}{2}} \), the following fourth-order one-sided approximation is incorporated:

\[
\frac{1}{24} \left( 26u'_{\frac{3}{2}} - 5u'_{\frac{5}{2}} + 4u'_{\frac{7}{2}} - u'_{\frac{9}{2}} \right) = \frac{1}{\Delta x_j} (u_2 - u_1).
\]

By symmetry, the derivative at the last node, \( u_{N+\frac{1}{2}} \), is given by:

\[
\frac{1}{24} \left( 26u'_{N+\frac{1}{2}} - 5u'_{N-\frac{1}{2}} + 4u'_{N-\frac{3}{2}} - u'_{N-\frac{5}{2}} \right) = \frac{1}{\Delta x_j} (u_{N+1} - u_N).
\]

In this implicit method, we assume that the values are known at nodes \( i = 1, ..., N+1 \), and we are trying to calculate the derivatives for the cell-centered nodes, which are at \( i = 3/2, ..., (2N + 1)/2 \). The formulation is shown by the following matrix equation.

\[
A \frac{\partial}{\partial x_j} \begin{bmatrix} u_{\frac{3}{2}} \\ \vdots \\ u_{\frac{2N+1}{2}} \end{bmatrix} = \frac{1}{\Delta x_j} \begin{bmatrix} u_2 - u_1 \\ \vdots \\ u_{N+1} - u_N \end{bmatrix},
\]
where

\[
A = \begin{bmatrix}
26 & -5 & 4 & -1 & \cdot & \cdot & 0 \\
1 & 22 & 1 & 0 & \cdot & \cdot & 0 \\
0 & 1 & 22 & 1 & 0 & \cdot & 0 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & \cdot & \cdot & 0 & 1 & 22 & 1 \\
0 & \cdot & \cdot & -1 & 4 & -5 & 26 \\
\end{bmatrix}.
\] (4.2.8)

Since all dimensions are known in advance, the matrix \( A \) can be calculated in advance.

Also, the matrix can be decomposed in advance using \( LU \)-decomposition, and the matrices \( L \) and \( U \) stored for use within the time-stepping loop. This provides greater computational efficiency since no inverses are necessary to be computed in every time step. Both of these schemes are simple to implement. You can find the partial derivative in any direction in 2-D or 3-D simply by applying the stencil in the correct direction in a one-dimensional fashion.

Xie et.al. developed an explicit fourth-order staggered FDTD method for Maxwell’s Equations that employs the same explicit fourth-order staggered scheme to calculate the spatial derivatives [69].

To account for those near-boundary nodes that Turkel’s fourth-order explicit scheme cannot calculate the derivatives for, we derived the following scheme to use for staggered grids to find the derivative of the near-boundary node 2. This is the case when the values at the half grid points \( i = 3/2, \ldots, (2N + 1)/2 \), and the derivatives at the grid points \( i = 1, \ldots, N + 1 \) are being computed. Turkel’s explicit stencil will only allow us to find the derivatives in this case at nodes \( i = 3, \ldots, N - 1 \), called internal nodes. For near-boundary node 2, we use

\[
u'_2 = \frac{1}{h} \left( au_3 + bu_5 + cu_7 + du_9 + eu_{11} \right), \quad (4.2.9)
\]
where the coefficients are found by matching the Taylor series expansion. We chose five nodes all together, including the one available to the left along with 4 known on the right. Five nodes provides us with a fourth-order explicit scheme. Solving the $5 \times 5$ system gives the coefficients

$$a = -\frac{22}{24} \quad b = \frac{17}{24} \quad c = \frac{9}{24} \quad d = -\frac{5}{24} \quad e = \frac{1}{24}.$$ (4.2.10)

It is well known that the scheme to find the derivative at the opposite near-boundary point $N$ would be given by

$$u'_N = \frac{1}{h} \left( -au_{N+\frac{1}{2}} - bu_{N-\frac{1}{2}} - cu_{N-\frac{3}{2}} - du_{N-\frac{5}{2}} - eu_{N-\frac{7}{2}} \right),$$ (4.2.11)

where the coefficients, $a$ through $e$ are the same as those given in equation (4.2.10)[30, 38].

Most of the time, boundary conditions are applied to nodes that lie directly on the boundary, so there is no need to calculate the derivative at those locations. However, we derived a scheme for a staggered grid, used to find the derivative of the nodes that lie external to the nodes at which values are known for, and which lie on the boundary of the domain. It is a one-sided explicit stencil to find the derivative at $u_1$, assuming that $u_1$ is the first node:

$$u'_1 = \frac{1}{h} \left( au_{\frac{3}{2}} + bu_{\frac{5}{2}} + cu_{\frac{7}{2}} + du_{\frac{9}{2}} + eu_{\frac{11}{2}} \right)$$ (4.2.12)

with coefficients

$$a = -\frac{31}{8} \quad b = \frac{229}{4} \quad c = -\frac{75}{8} \quad d = \frac{37}{8} \quad e = -\frac{11}{12}.$$ (4.2.13)

And, for boundary point $N + 1$, we will use

$$u'_{N+1} = \frac{1}{h} \left( -au_{N+\frac{1}{2}} - bu_{N-\frac{1}{2}} - cu_{N-\frac{3}{2}} - du_{N-\frac{5}{2}} - eu_{N-\frac{7}{2}} \right),$$ (4.2.14)
where the coefficients are the same as in equation (4.2.13).

We performed a test of convergence rates using Turkel’s fourth-order explicit method for internal nodes, along with our stencils to calculate the derivatives at the near-boundary and boundary points. These results are shown in Table 4.1 and Table 4.2.

4.3 High Order Temporal Derivatives for Staggered Grids

If one is to use high-order spatial derivatives, it does not make much sense to continue to employ the second-order time derivatives from the original Yee algorithm. This is because the accumulated time errors will dominate the overall error, unless you use a very small time step, but then this will make your scheme less efficient than if a higher order scheme in space were implemented. Turkel and Yefet employ both second-order time marching as well as fourth-order time marching along with their fourth-order spacial derivatives, and they deemed these to be Ty(2, 4) and the Ty(4, 4) schemes, respectively [60]. Their computational results clearly show that for the same size mesh, the time steps required for the Ty(2, 4) scheme are at least 4 times smaller than the time steps required for the Ty(4, 4) scheme. Later, Turkel derives a different fourth-order time marching scheme that involves a backward finite difference stencil in time that requires storage of values for three previous time steps [59]. This is comparable to Xie et.al.’s fourth-order FDTD method, which also employs fourth-order explicit time stepping that employs a backward difference approximation, thus requiring the history values of all fields at four time levels [68]. We found a more appealing time-stepping scheme that is based on the classical fourth-order explicit
Runge-Kutta (RK) method, but which employs time staggering [22]. It is considered to be a variation of the RK family of time integrators to solve systems of linear wave equations on uniform, time-staggered grids. They are called staggered RK methods [22].

When an ODE is written in the form

\begin{align*}
  u' &= f(t, v(t)), \\
  v' &= g(t, u(t)),
\end{align*}

where \( u \) and \( v \) each could be vectors, it allows the quantities to be staggered spatially.

The most general form of the staggered Runge-Kutta (RK) method (RKS) is the following s-stage formula with \( k = \Delta t \) time step:

\begin{align*}
  d_1 &= k f \left( t_{n+\frac{1}{2}}^+, v_{n+\frac{1}{2}}^+ \right), \\
  d_1 &= k g \left( t_n + c_2 k, u_n + a_{21} d_1 \right), \\
  d_3 &= k f \left( t_{n+\frac{1}{2}}^+ + c_3 k, v_{n+\frac{1}{2}}^+ + a_{32} d_2 \right), \\
  d_4 &= k g \left( t_n + c_4 k, u_n + a_{41} d_1 + a_{43} d_3 \right), \\
  &\vdots \\
  d_s &= k f \left( t_{n+\frac{1}{2}}^+ + c_s k, v_{n+\frac{1}{2}}^+ + a_{s2} d_2 + \cdots + a_{s,s-1} d_{s-1} \right), \\
  u_{n+1} &= u_n + b_1 d_1 + b_3 d_3 + \cdots + b_s d_s 
\end{align*}

if \( s \) is odd [22]. To advance \( v \), the same formula is used, but with \( f \) and \( g \) reversed and \( 1/2 \) added to all time levels. In comparison with other fourth-order time schemes, Ghrist et. al. named their most appealing scheme RKS4, which is given by:

\begin{align*}
  d_1 &= k f \left( t_{n+\frac{1}{2}}^+, v_{n+\frac{1}{2}}^+ \right), \\
  d_2 &= k g \left( t_n, u_n \right), \\
  d_3 &= k f \left( t_{n+\frac{1}{2}}^+ - k, v_{n+\frac{1}{2}}^+ - d_2 \right), \\
  d_4 &= k g \left( t_n + k, u_n + d_1 \right), \\
  d_5 &= k f \left( t_{n+\frac{1}{2}}^+ + k, v_{n+\frac{1}{2}}^+ + d_4 \right), \\
  u_{n+1} &= u_n + \frac{11}{12} d_1 + \frac{1}{24} d_3 + \frac{1}{24} d_5 
\end{align*}

While it appears there are five stages, the \( d_1 \) stage is identical to the \( d_2 \) stage that is needed in calculation of \( v_{n+\frac{3}{2}} \) later, which results in computation of only four steps.
If efficient storage is utilized. As before, to calculate $v_{n+\frac{1}{2}}$, the same procedure is used with the exception of $f$ and $g$ reversed, and all time steps incremented by $1/2$.

J.G. Verwer compared the RKS4, which he refers to as StaggeredLF4 (LF for “leap frog”), with a symmetric-composition method derived from symplectic Euler, called SymmetricCO4, also a fourth-order explicit time-stepping scheme. However, in the SymmetricCO4 method, time is not staggered. Verwer claims that the SymmetricCO4 method appears to be as efficient, if not more, than the staggered time method, but needs to do more investigation on this claim [63]. Later, in an unpublished note [64], Verwer investigated more and compared the StaggeredLF4, SymmetricCO4, and two explicit, non-staggered Runge-Kutta-Nyström (RKN) methods. For the RKN methods, he chose to use a fourth-order, five-stage method, called RKN45, as well as a fifth-order, seven-stage method called RKN57. In Verwer’s unpublished paper, he finds that the RKN45 and RKN57 outperform the Staggered and Symmetric methods when solving the Sine-Gordon equation. However, it is noted that the StaggeredLF4 method ”has a larger scaled-stability interval and suffers less from order reduction, at least in the limit to convergence”. There is more work that can be done to investigate these different time marching schemes, especially in metamaterials and cloaking FDTD simulations.

4.4 Numerical Results

First, we tested the convergence rates of the spatially staggered derivatives. To our knowledge, nobody has coupled these fourth-order explicit staggered schemes together that we use here. We combined Turkel’s explicit fourth-order scheme along with our
explicit schemes to compute the derivatives of the near-boundary and boundary nodes, which have similar forms to those schemes Lele derived. We examine the case when the values at the half nodes are known, at \( u_{\frac{3}{2}}, ..., u_{\frac{2N+1}{2}} \), and we wish to calculate the values of the derivatives of \( u \) at the locations \( i = 1, ..., N+1 \). One can see that Turkel’s explicit scheme can only help us find the derivatives at the nodes at \( i = 3, ..., N - 1 \). So, we use our scheme to calculate the derivatives at the near-boundary nodes \( i = 2, N \) as well as the boundary nodes \( i = 1, N + 1 \).

The first analytic solution we consider is very smooth. We let

\[ E(x, y) = \sin(\pi x) \cos(\pi y) \] in \([0, 1] \times [0, 1]\). Using this scheme to calculate partials in both directions, \( \frac{\partial E}{\partial x} \) and \( \frac{\partial E}{\partial y} \), we see the convergence rates in Table 4.1 meet or exceed fourth-order for such a smooth function.

**Table 4.1.** \( L^\infty \) errors for mesh of size \( n \times n \) with solution \( \sin(\pi x) \cos(\pi y) \)

<table>
<thead>
<tr>
<th>n</th>
<th>Max ( E_x ) Error</th>
<th>Rate</th>
<th>Max ( E_y ) Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0299</td>
<td></td>
<td>0.0261</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.0019</td>
<td>3.967</td>
<td>6.789e-4</td>
<td>5.265</td>
</tr>
<tr>
<td>40</td>
<td>1.146e-4</td>
<td>4.051</td>
<td>1.898e-5</td>
<td>5.158</td>
</tr>
<tr>
<td>80</td>
<td>6.885e-6</td>
<td>4.057</td>
<td>5.590e-7</td>
<td>5.085</td>
</tr>
<tr>
<td>160</td>
<td>4.208e-7</td>
<td>4.032</td>
<td>1.694e-8</td>
<td>5.044</td>
</tr>
<tr>
<td>320</td>
<td>2.599e-8</td>
<td>4.039</td>
<td>5.213e-10</td>
<td>5.022</td>
</tr>
<tr>
<td>640</td>
<td>1.614e-9</td>
<td>4.009</td>
<td>1.721e-11</td>
<td>4.921</td>
</tr>
<tr>
<td>1280</td>
<td>1.020e-10</td>
<td>3.984</td>
<td>3.597e-12</td>
<td>2.258</td>
</tr>
</tbody>
</table>

The last analytic solution we consider is \( E(x, y) = \exp(4xy) + 2x - 4xy \) in the same domain, which is not as smooth as the previous example. In this case, the errors were identical for the partials with respect to both \( x \) and \( y \), so we list only one column of errors in Table 4.2. We see the convergence rates are converging towards four as...
the mesh is refined, as we should.

<table>
<thead>
<tr>
<th>n</th>
<th>Max $E_x/E_y$ Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.984</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.197</td>
<td>3.332</td>
</tr>
<tr>
<td>40</td>
<td>0.015</td>
<td>3.715</td>
</tr>
<tr>
<td>80</td>
<td>.001</td>
<td>3.907</td>
</tr>
<tr>
<td>160</td>
<td>7.109e-5</td>
<td>3.814</td>
</tr>
<tr>
<td>320</td>
<td>4.565e-6</td>
<td>3.961</td>
</tr>
<tr>
<td>640</td>
<td>2.891e-7</td>
<td>3.981</td>
</tr>
<tr>
<td>1280</td>
<td>1.821e-8</td>
<td>3.989</td>
</tr>
</tbody>
</table>

Now, to test our own fourth-order FDTD method for normalized, homogeneous Maxwell’s Equations in free space, consider the partitioned ODE system

$$u' = f(t,v), \quad v' = g(t,u),$$

and using $u = \vec{E}$, $v = \vec{H}$, $f = \nabla \times$, $g = -\nabla \times$, and Ampere’s and Faraday’s laws with $\epsilon_0 = \mu_0 = 1$, we have

$$\frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H},$$
$$\frac{\partial \vec{H}}{\partial t} = -\nabla \times \vec{E},$$

on the domain $[0,1] \times [0,1]$.

Here, we are using the conducting boundary conditions $\hat{n} \times \vec{E} = 0$ on the boundary. Therefore, $E_x(x,0) = E_x(x,1) = 0$ and $E_y(0,y) = E_y(1,y) = 0$, which come naturally from the analytical solutions. If we consider the $z$-polarized $TE_z$ modes,
the component equations are

\[
\frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y}, \quad \frac{\partial E_y}{\partial t} = -\frac{\partial H_z}{\partial x}, \quad \frac{\partial H_z}{\partial t} = \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}.
\]  

(4.4.3)

We use the 2-D grid shown in Figure 4.2 for our simulations.

![Two-dimensional staggered grid used for TE₂ mode simulations](image)

**Figure 4.2.** Two-dimensional staggered grid used for TE₂ mode simulations

We will be using the following known analytic solution in order to check convergence rates:

\[
E_x = \frac{\pi}{\sqrt{2}} \cos (\pi x) \sin (\pi y) \sin (\omega t) \quad (4.4.4)
\]

\[
E_y = -\frac{\pi}{\sqrt{2}} \sin (\pi x) \cos (pi y) \sin (\omega t) \quad (4.4.5)
\]

\[
H_z = -\pi \cos (\pi x) \cos (\pi y) \cos (\omega t) \quad (4.4.6)
\]

with \( \omega = \sqrt{2}\pi \) on the domain \([0, 1] \times [0, 1]\)
Using the same notation as in [58], the 2-D explicit fourth-order discretized scheme is given by

\[
\begin{align*}
\frac{\partial E_x}{\partial t}\bigg|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{\partial H_z}{\partial y}\bigg|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}, \\
\frac{\partial E_y}{\partial t}\bigg|_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} &= -\frac{\partial H_z}{\partial x}\bigg|_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}}, \\
\frac{\partial H_z}{\partial t}\bigg|_{i,j+1}^{n+1} &= \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\bigg|_{i,j+1}^{n+1}.
\end{align*}
\] (4.4.7)

To our knowledge, nobody else has implemented this combination of schemes to simulate Maxwell’s Equations.

The errors for the fields \(E_x\) and \(E_y\) were always the same, and yielded continuously lower errors than the \(H_z\) field, as you can see in Tables 4.3 - 4.7. We ran the simulations with \(\Delta x = \Delta y\) on domain \([0,1] \times [0,1]\) and time step \(\Delta t = CFL\Delta x\), using various \(CFL\) values ranging from 0.5 down to 0.005.

In Tables 4.3 - 4.5 we used \(\Delta x\) values of \(1/10, 1/20, 1/40, \) and \(1/80\). We can see that the \(CFL\) value being used doesn’t seem to be affecting the convergence rates. Also, although the convergence rates for the electric field seems to be consistently converging towards four, as it should, the convergence rates of the magnetic field are a little bit jumpy. We see this problem eliminated in Tables 4.6 and 4.7 when the values of \(\Delta x\) are reduced to \(1/40, 1/80, 1/160, 1/320\). Here, both the electric and magnetic fields converge towards order four, regardless of the \(CFL\) value, as before.
Table 4.3. $L^\infty$ errors for $CFL = 0.5$ and $\Delta x$ from $1/10$ to $1/80$

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Max $E_x$ Error</th>
<th>Rate</th>
<th>Max $H_z$ Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>2.41e-4</td>
<td></td>
<td>6.81e-4</td>
<td></td>
</tr>
<tr>
<td>1/20</td>
<td>9.5e-6</td>
<td>4.665</td>
<td>9.36e-6</td>
<td>6.185</td>
</tr>
<tr>
<td>1/40</td>
<td>5.01e-7</td>
<td>4.245</td>
<td>1.5e-6</td>
<td>2.642</td>
</tr>
<tr>
<td>1/80</td>
<td>3.08e-8</td>
<td>4.024</td>
<td>1.23e-7</td>
<td>3.608</td>
</tr>
</tbody>
</table>

Table 4.4. $L^\infty$ errors for $CFL = 0.1$ and $\Delta x$ from $1/10$ to $1/80$

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Max $E_x$ Error</th>
<th>Rate</th>
<th>Max $H_z$ Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>2.34e-4</td>
<td></td>
<td>6.73e-4</td>
<td></td>
</tr>
<tr>
<td>1/20</td>
<td>9.22e-6</td>
<td>4.666</td>
<td>7.81e-6</td>
<td>6.429</td>
</tr>
<tr>
<td>1/40</td>
<td>4.7e-7</td>
<td>4.294</td>
<td>1.42e-6</td>
<td>2.459</td>
</tr>
<tr>
<td>1/80</td>
<td>2.91e-8</td>
<td>4.013</td>
<td>1.19e-7</td>
<td>3.577</td>
</tr>
</tbody>
</table>

Table 4.5. $L^\infty$ errors for $CFL = 0.005$ and $\Delta x$ from $1/10$ to $1/80$

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Max $E_x$ Error</th>
<th>Rate</th>
<th>Max $H_z$ Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>2.33e-4</td>
<td></td>
<td>6.70e-4</td>
<td></td>
</tr>
<tr>
<td>1/20</td>
<td>9.17e-6</td>
<td>4.667</td>
<td>7.70e-6</td>
<td>6.443</td>
</tr>
<tr>
<td>1/40</td>
<td>4.66e-7</td>
<td>4.299</td>
<td>1.41e-6</td>
<td>2.449</td>
</tr>
<tr>
<td>1/80</td>
<td>2.89e-8</td>
<td>4.011</td>
<td>1.19e-7</td>
<td>3.567</td>
</tr>
</tbody>
</table>
Table 4.6. $L^\infty$ errors for $CFL = 0.5$ and $\Delta x$ from 1/40 to 1/320

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Max $E_x$ Error</th>
<th>Rate</th>
<th>Max $H_z$ Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/40</td>
<td>5.01e-7</td>
<td></td>
<td>1.50e-6</td>
<td></td>
</tr>
<tr>
<td>1/80</td>
<td>3.08e-8</td>
<td>4.024</td>
<td>1.23e-7</td>
<td>3.608</td>
</tr>
<tr>
<td>1/160</td>
<td>1.90e-9</td>
<td>4.019</td>
<td>8.66e-9</td>
<td>3.828</td>
</tr>
<tr>
<td>1/320</td>
<td>1.18e-10</td>
<td>4.009</td>
<td>5.72e-10</td>
<td>3.902</td>
</tr>
</tbody>
</table>

Table 4.7. $L^\infty$ errors for $CFL = 0.005$ and $\Delta x$ from 1/40 to 1/320

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>Max $E_x$ Error</th>
<th>Rate</th>
<th>Max $H_z$ Error</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/40</td>
<td>4.66e-7</td>
<td></td>
<td>1.41e-6</td>
<td></td>
</tr>
<tr>
<td>1/80</td>
<td>2.98e-8</td>
<td>3.967</td>
<td>1.19e-7</td>
<td>3.567</td>
</tr>
<tr>
<td>1/160</td>
<td>1.81e-9</td>
<td>4.041</td>
<td>8.38e-9</td>
<td>3.828</td>
</tr>
<tr>
<td>1/320</td>
<td>1.14e-10</td>
<td>3.989</td>
<td>5.55e-10</td>
<td>3.916</td>
</tr>
</tbody>
</table>
CHAPTER 5

TIME DOMAIN DISCONTINUOUS GALERKIN METHOD FOR METAMATERIALS

5.1 Introduction to Discontinuous Galerkin Method

As early as 1971, discontinuous basis functions and penalty terms were used to approximate second-order elliptic equations with Dirichlet boundary conditions. In 1973, Babuška et. al. [4, 5, 6] suggested an application of the penalty method to the finite element method. In that same year, Reed and Hill introduced the discontinuous Galerkin (DG) method for solving linear hyperbolic equations [52]. Then, in 1974 a \textit{a priori} error estimates for two-dimensional linear hyperbolic equations solved by using the DG method were derived by Lesaint and Raviart [32]. Unfortunately, the ideas and theory were there, but the computational power necessary to implement these methods was not. Therefore, these methods were overlooked for many years. Since the 1990s, DG methods have become an attractive research topic due to their many benefits.

As opposed to the finite difference methods, in which the convergence rates of the schemes generally depend on the order of the Taylor expansion being used to approximate the derivatives, solutions in the DG method are approximated using basis functions, as in $u(x) \approx \sum_{i=1}^{N} w_i \phi_i(x)$. Hence, the order of accuracy simply depends on the number of basis functions in the expansion. Usually a nodal method is employed, with the nodes lying on each triangular element as shown in Figure 5.1. The information used to calculate the solution on each element only involves communication with directly neighboring elements and not all elements in the mesh, unlike other
finite element methods. Each element shares flux information across the connecting edge. This flux is continuous and can be extended to a continuous flux on the entire domain. The flux information can be see in Figure 5.2. The DG algorithms have less numerical diffusion, which provides support for more accurate solutions compared to other methods. Since solutions are solved in a discontinuous function space, the method captures discontinuity better. This method is very flexible in that it is simpler to implement $h-p$ adaptive and non-conforming meshes with it. The methods are easy to implement, can be coupled with conforming methods, and the code is generally shorter and more efficient. Finally, it lends itself to be easily used with parallelization.

Figure 5.1. A typical triangular element used in the DG method. Courtesy of Stanford VLF Group

Due to many of the benefits and flexibilities listed above, the DG method has gained more popularity in recent years in solving all types of differential equations.
Recently, over the past decade, interest has grown considerably in developing DG methods to solve Maxwell’s equations [11, 13, 17, 24, 25, 27, 44]. During that time, work has been done to develop DG methods to solve Maxwell’s Equations in dispersive media [42, 28], whose permittivity depends on the wave frequency. However, the study of DG method for Maxwell’s equations in complex media such as metamaterials is quite limited.

5.2 Developing the DG Method for a Metamaterial Model

In this section, we extend the DG technique developed by [25] to the metamaterial model. The original development was discussed in [35].

To make the derivation simple, we assume that $\epsilon_0 = 1$ and $\mu_0 = 1$ in the governing
equations [33]:

\[ \frac{\partial E}{\partial t} = \nabla \times H - J, \quad \text{in } \Omega \times (0,T), \quad (5.2.1) \]

\[ \frac{\partial H}{\partial t} = -\nabla \times E - K, \quad \text{in } \Omega \times (0,T), \quad (5.2.2) \]

\[ \frac{\partial J}{\partial t} + \Gamma_e J = \epsilon_0 \omega^2_{pe} E, \quad \text{in } \Omega \times (0,T), \quad (5.2.3) \]

\[ \frac{\partial K}{\partial t} + \Gamma_m K = \mu_0 \omega^2_{pm} H, \quad \text{in } \Omega \times (0,T), \quad (5.2.4) \]

Let us rewrite (5.2.1) and (5.2.2) with added source functions \( f \) and \( g \) in conservation form

\[ \frac{\partial \tilde{q}}{\partial t} + \nabla \cdot \tilde{f}(\tilde{q}) = \tilde{S}, \quad (5.2.5) \]

where we denote

\[ \tilde{q} = \begin{bmatrix} \tilde{E} \\ \tilde{H} \end{bmatrix}, \quad \tilde{S} = \begin{bmatrix} S_E \\ S_H \end{bmatrix} = \begin{bmatrix} -J + \tilde{f} \\ -K + \tilde{g} \end{bmatrix}, \quad \tilde{f}(\tilde{q}) = \begin{bmatrix} -e_i \times \tilde{H} \\ e_i \times \tilde{E} \end{bmatrix}, \]

and \( \tilde{f}(\tilde{q}) = [F_1(\tilde{q}), F_2(\tilde{q}), F_3(\tilde{q})]^T \). Here \( e_i \) are the three Cartesian unit vectors.

We assume that the domain \( \Omega \) is decomposed into tetrahedral (or triangular in 2-D) elements \( \Omega_k \), and the numerical solution \( \tilde{q}_N \) is represented as

\[ \tilde{q}_N(\vec{x},t) = \sum_{j=1}^{N_n} \tilde{q}_j(\vec{x}_j, t)L_j(\vec{x}) = \sum_{j=1}^{N_n} \tilde{q}_j(t)L_j(\vec{x}), \quad (5.2.6) \]

where \( L_j(\vec{x}) \) is the multivariate Lagrange interpolation polynomial of degree \( n \). Here \( N_n = \frac{1}{6}(n+1)(n+2)(n+3) \) in 3-D; while \( N_n = \frac{1}{2}(n+1)(n+2) \) in 2-D.

Multiplying (5.2.5) by a test function \( L_i(\vec{x}) \) and integrating over each element \( \Omega_k \), we obtain

\[ \int_{\Omega_k} \left( \frac{\partial \tilde{q}_N}{\partial t} + \nabla \cdot \tilde{f}(\tilde{q}_N) - \tilde{S}_N \right)L_i(\vec{x})d\vec{x} = \int_{\partial \Omega_k} \hat{n} \cdot (\tilde{f}(\tilde{q}_N) - \tilde{f}_N)L_i(\vec{x})d\vec{x}, \quad (5.2.7) \]
where \( \hat{n} \) is an outward normal unit vector of \( \partial \Omega_k \), and \( \vec{f}_N^\ast \) is a numerical flux. For the Maxwell’s equations, we usually choose the upwind flux \[25\]

\[
\hat{n} \cdot (\vec{f}(\vec{q}_N) - \vec{f}_N^\ast) = \begin{cases} 
\frac{1}{2} \hat{n} \times ([\vec{H}_N] - \hat{n} \times [\vec{E}_N]) \\
\frac{1}{2} \hat{n} \times (-\hat{n} \times [\vec{H}_N] - [\vec{E}_N])
\end{cases},
\]

where \([\vec{E}_N] = \vec{E}^+ - \vec{E}^-\), and \([\vec{H}_N] = \vec{H}^+ - \vec{H}^-\). Here superscripts '+' and '-' refer to field values from the neighbor element and the local element itself, respectively.

Substituting (5.2.6) into (5.2.7), we obtain the elementwise equations for the electric field components

\[
\sum_{j=0}^{N} (M_{ij} \frac{d\vec{E}_j}{dt} - S_{ij} \times \vec{H}_j - M_{ij} S_{E,j}) = \frac{1}{2} \sum_l F_{il} \cdot \hat{n}_l \times (\vec{H}_l - \hat{n}_l \times \vec{E}_l),
\]

(5.2.8)

and for the magnetic field components

\[
\sum_{j=0}^{N} (M_{ij} \frac{d\vec{H}_j}{dt} + S_{ij} \times \vec{E}_j - M_{ij} S_{H,j}) = \frac{1}{2} \sum_l F_{il} \cdot \hat{n}_l \times (-\hat{n}_l \times \vec{H}_l - \vec{E}_l),
\]

(5.2.9)

where

\[
M_{ij} = (L_i(\vec{x}), L_j(\vec{x}))_{\Omega_k}, \quad S_{ij} = (L_i(\vec{x}), \nabla L_j(\vec{x}))_{\Omega_k}
\]

represent the local mass and stiffness matrices, respectively. Furthermore,

\[
F_{il} = (L_i(\vec{x}), L_l(\vec{x}))_{\partial \Omega_k}
\]

represents the face-based mass matrix.

We can rewrite (5.2.8)-(5.2.9) in a fully explicit form, while the constitutive equations (5.2.3)-(5.2.4) keep the same form. In summary, we have the following semi-
discrete Discontinuous Galerkin scheme:

\[
\frac{d\vec{E}_N}{dt} = M^{-1} S \times \vec{H}_N - \vec{J}_N + \vec{f}_N + \frac{1}{2} M^{-1} F\left(\hat{n} \times ([\vec{H}_N] - \hat{n} \times [\vec{E}_N])\right)\big|_{\partial\Omega_h},
\]
(5.2.10)

\[
\frac{d\vec{H}_N}{dt} = -M^{-1} S \times \vec{E}_N - \vec{K}_N + \vec{g}_N - \frac{1}{2} M^{-1} F\left(\hat{n} \times (\hat{n} \times [\vec{H}_N] + [\vec{E}_N])\right)\big|_{\partial\Omega_h},
\]
(5.2.11)

\[
\frac{d\vec{J}_N}{dt} = \omega_e^2 \vec{E}_N - \Gamma_e \vec{J}_N,
\]
(5.2.12)

\[
\frac{d\vec{K}_N}{dt} = \omega_m^2 \vec{H}_N - \Gamma_m \vec{K}_N.
\]
(5.2.13)

The system (5.2.10)-(5.2.13) can be solved by various methods developed for a system of ordinary differential equations

\[
\frac{d\bar{u}_h}{dt} = \mathcal{L}(\bar{u}_h, t),
\]
(5.2.14)

where \(\bar{u}_h\) is the vector of unknowns. In our implementation we adopt the classic low-storage five-stage fourth-order explicit Runge-Kutta method [26, §3.4]:

\[
\bar{p}^{(0)} = \bar{u}_h^{(0)}, \quad \bar{r}^{(0)} = 0,
\]

\[
 i \in [1, 5]: \left\{ \begin{array}{l}
 \bar{r}^{(i)} = a_i \bar{r}^{(i-1)} + \tau \mathcal{L}(\bar{p}^{(i-1)}, n\tau + c_i\tau), \\
 \bar{p}^{(i)} = \bar{p}^{(i-1)} + b_i \bar{r}^{(i)}, \\
 \end{array} \right.
\]

\[
\bar{u}_h^{n+1} = \bar{p}^{(5)},
\]

where coefficients \(a_i, b_i\) and \(c_i\) are fixed constants given in Table 3.2 of [26].
5.3 Numerical Results

Here, we consider the 2-D transverse magnetic metamaterial model in component form with $\omega^2 = \omega^2 = \Gamma_e = \Gamma_m = 1$:

\[
\begin{align*}
\frac{\partial H_x}{\partial t} &= -\frac{\partial E_z}{\partial y} - K_x + g_x \quad (5.3.1) \\
\frac{\partial H_y}{\partial t} &= \frac{\partial E_z}{\partial x} - K_y + g_y \quad (5.3.2) \\
\frac{\partial E_z}{\partial t} &= \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - J_z + f \quad (5.3.3) \\
\frac{\partial J_z}{\partial t} &= E_z - J_z \quad (5.3.4) \\
\frac{\partial K_x}{\partial t} &= H_x - K_x \quad (5.3.5) \\
\frac{\partial K_y}{\partial t} &= H_y - K_y. \quad (5.3.6)
\end{align*}
\]

We consider a problem with an exact solution in order to check the convergence rate obtained by our implementation. We solve the above system and compare it using the following set of exact solutions on domain $\Omega = (0, 1)^2$:

\[
\begin{align*}
\vec{H} &\equiv \begin{pmatrix} H_x \\ H_y \end{pmatrix} = \begin{pmatrix} y \sin(\pi x) \\ -\frac{1}{2} y^2 \pi \cos(\pi x) \end{pmatrix} t \quad (5.3.7) \\
E_z &= y(1 - y) \sin(\pi x) t \\ (5.3.8) \\
J_z &= [t - (1 - e^{-t})] y(1 - y) \sin(\pi x) \\ (5.3.9) \\
\vec{K} &\equiv \begin{pmatrix} K_x \\ K_y \end{pmatrix} = \begin{pmatrix} y \sin(\pi x) \\ -\frac{1}{2} y^2 \pi \cos(\pi) \end{pmatrix} \left[ t - (1 - e^{-t}) \right] \\ (5.3.10) \\
f &= y(1 - y) \sin(\pi x) - \frac{1}{2} y^2 \pi^2 \sin(\pi x) t + \cdots \\
&\quad \sin(\pi x) t + \left[ t - (1 - e^{-t}) \right] y(1 - y) \sin(\pi x) \\ (5.3.11) \\
g_x &= y \sin(\pi x) + (1 - 2y) \sin(\pi x) t + y \sin(\pi x) \left[ t - (1 - e^{-t}) \right] \\ (5.3.12) \\
g_y &= -\frac{1}{2} y^2 \pi \cos(\pi x) - y(1 - y) \pi \cos(\pi x) t - \frac{1}{2} y^2 \pi \cos(\pi x) \left[ t - (1 - e^{-t}) \right] \\ (5.3.13)
\end{align*}
\]
Tables 5.1-5.3 show the results for first order elements on meshes of size $nx \times nx$, where $nx = 5, 10, 20, 40, 80$.

**Table 5.1.** $L^\infty$ errors with $\tau = 10^{-5}$ for 10 time steps

<table>
<thead>
<tr>
<th>Errors</th>
<th>$H_x$</th>
<th>$H_y$</th>
<th>$E_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.6469e-009</td>
<td>2.3492e-009</td>
<td>1.0273e-008</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.5213e-009</td>
<td>1.3908e-009</td>
<td>6.9340e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.799</td>
<td>0.756</td>
<td>0.567</td>
</tr>
<tr>
<td>20</td>
<td>9.7608e-010</td>
<td>7.4566e-010</td>
<td>4.4345e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.640</td>
<td>0.899</td>
<td>0.645</td>
</tr>
<tr>
<td>40</td>
<td>6.9603e-010</td>
<td>4.9341e-010</td>
<td>3.1976e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.488</td>
<td>0.596</td>
<td>0.472</td>
</tr>
<tr>
<td>80</td>
<td>5.7662e-010</td>
<td>4.1999e-010</td>
<td>2.5944e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.272</td>
<td>0.232</td>
<td>0.302</td>
</tr>
</tbody>
</table>

**Table 5.2.** $L^\infty$ errors with $\tau = 10^{-6}$ for 100 time steps

<table>
<thead>
<tr>
<th>Errors</th>
<th>$H_x$</th>
<th>$H_y$</th>
<th>$E_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.9127e-009</td>
<td>2.4873e-009</td>
<td>9.5777e-009</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.5322e-009</td>
<td>1.3985e-009</td>
<td>5.8384e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.927</td>
<td>0.831</td>
<td>0.714</td>
</tr>
<tr>
<td>20</td>
<td>7.8899e-010</td>
<td>7.4495e-010</td>
<td>3.1112e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.958</td>
<td>0.909</td>
<td>0.908</td>
</tr>
<tr>
<td>40</td>
<td>4.0992e-010</td>
<td>3.8394e-010</td>
<td>1.6740e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.945</td>
<td>0.956</td>
<td>0.894</td>
</tr>
<tr>
<td>80</td>
<td>2.1795e-010</td>
<td>1.9553e-010</td>
<td>9.2835e-010</td>
</tr>
<tr>
<td>Rate</td>
<td>0.911</td>
<td>0.973</td>
<td>0.851</td>
</tr>
</tbody>
</table>
Table 5.3. $L^\infty$ errors with $\tau = 10^{-7}$ for 1000 time steps

<table>
<thead>
<tr>
<th>Errors</th>
<th>$H_x$</th>
<th>$H_y$</th>
<th>$E_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2.9393e-009</td>
<td>2.5099e-009</td>
<td>9.6818e-009</td>
</tr>
<tr>
<td>Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.5463e-009</td>
<td>1.4113e-009</td>
<td>5.8589e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.927</td>
<td>0.831</td>
<td>0.725</td>
</tr>
<tr>
<td>20</td>
<td>8.0319e-010</td>
<td>7.4493e-010</td>
<td>3.0168e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.945</td>
<td>0.922</td>
<td>0.958</td>
</tr>
<tr>
<td>40</td>
<td>4.0788e-010</td>
<td>3.8391e-010</td>
<td>1.5371e-009</td>
</tr>
<tr>
<td>Rate</td>
<td>0.978</td>
<td>0.956</td>
<td>0.973</td>
</tr>
<tr>
<td>80</td>
<td>2.0583e-010</td>
<td>1.9551e-010</td>
<td>7.8310e-010</td>
</tr>
<tr>
<td>Rate</td>
<td>0.987</td>
<td>0.974</td>
<td>0.973</td>
</tr>
</tbody>
</table>
CHAPTER 6
CONCLUSIONS AND FUTURE WORK

6.1 Summary

This thesis has focused on reviewing, implementing, and creating new schemes for dispersive metamaterial models.

In Chapter 2, we go back to basics. Beginning with Ampere’s and Faraday’s Laws, we looked at the constitutive equations and how they relate the electromagnetic (EM) fields to the material parameters, permeability and permittivity. We discussed the Drude model and other competing models, and showed how to go from the frequency to the time domain, eventually building up to the governing equations for double negative (DNG) or negative-index metamaterial (NIM) simulation. For this particular set of equations, we showed existence, uniqueness, as well as the divergence free conditions via Lemmas. The classic FDTD model and algorithm is introduced, along with how it is used with the PDEs that arise from the component forms of the EM fields. Within the FDTD method, we looked at the original second-order central differences that it employed, along with numerical dispersion and stability and some common absorbing boundary conditions (ABCs). Finally, we discretized our governing equations and showed the creation of a new dispersive FDTD method for metamaterials. We ran numerical simulations to show that the physics we observed matches that shown in several engineering papers. We applied this new method to several different examples, from slabs to cylinders, both small and large.

Chapter 3 gives readers an introduction to cloaking metamaterials. We discussed a few of the seminal works that started the now blossoming field of research in meta-
materials and cloaking. We discussed the linear transformation used to redirect the electromagnetic waves around the object to be hidden. We also describe the material parameters, which become radially dependent, making simulation of cloaks more challenging than the DNG metamaterials we looked at in Chapter 2. We showed several modern developments in the field of cloaking, which is just beginning to sprout. There are several applications that the technology that is being used in cloaking research will be able to be applied to in the future. We discussed the axisymmetric cylindrical cloak, which has radially-dependent material parameters. Permittivity is a tensor instead of a constant or simply a function, so a new method for solving Maxwell’s Equations for cloaks is developed and discussed. Numerical simulations were run that show results comparable to those seen in technical papers.

High order compact schemes were discussed in Chapter 4. We discussed several different spatial derivative schemes for staggered grids, and noted that some were quite common. We discussed both explicit and implicit fourth-order staggered grid schemes. We explained how nodes are chosen in order to develop compact schemes, and how to determine the coefficients and order. After discussing spatial derivatives, we discussed temporal derivatives for staggered time grids. The classical Runge-Kutta method is not stable for a short number of time steps, but we reviewed the staggered Runge-Kutta method, and showed how it is stable for shorter number of time steps, and our results showed excellent convergence rates and very small errors.

The discontinuous Galerkin (DG) method was discussed in Chapter 5. Background information was provided, which showed mostly benefits to using the DG method over other methods. It is shown to be flexible, stable, and powerful. We developed a DG
method based on the governing equations given in Chapter 2. A 2-D test case with known analytic solutions is run, and the data supports that errors are very low using this method.

6.2 Future Work

Since the late 1990s and the realizations of DNG metamaterials, research is booming in all areas of development, including simulations. Whenever a laboratory experiment is done, the need for mathematical and physical proof, and analysis arises. There are now four top methods that are being used to simulate electromagnetic wave propagation through metamaterials. These are FDTD, FEM, DG, and COMSOL Multiphysics™ package. Due to its ease of implementation and capability of tackling a wide range of problems, the FDTD method is still widely being used, and will continue being used in the future. There are several nice higher order finite difference schemes, as we showed in this thesis, but they have yet to be incorporated into simulations of metamaterials. We touched on this a bit in Chapter 4, but were unable to include results due to time constraints. We are planning to develop a fourth-order metamaterial and cloak scheme, which should be more accurate and efficient than the traditional Yee scheme. Stability and error analyses also need to be done for these higher order schemes. Another novel idea is to create hybrid schemes and possibly incorporating “mesh-free” methods, which have yet to be considered for metamaterial simulation to our knowledge. The FDTD scheme could easily be linked together with “meshless” methods via domain decomposition. We would also like to assess and expand the high order staggered time schemes discussed in Chapter 4. Along with
developing new FDTD schemes, it is also necessary to develop absorbing boundary conditions, such as a new PML, that will be compatible with the new scheme being used in the main computational domain.
APPENDIX A

EXCERPTS OF DNG METAMATERIAL MATLAB CODE

A.1 Fundamental Constants

cc=2.99792458e8; \textit{%} speed of light in free space
muz=4.0*pi*1.0e-7; \textit{%} permeability of free space
epsz=1.0/(cc*cc*muz); \textit{%} permittivity of free space
freq=3.0e10; \textit{%} center frequency of source excitation
lambda=cc/freq; \textit{%} center wavelength of source excitation
gamma=10^-8; \textit{%} \gamma_e=gamma_m=gamma number from p667 of paper
\texttt{wpem}=2*pi*sqrt(2)*freq; \textit{%} n=-1 case, should be 2.66573e11
\texttt{wpem}=2*pi*sqrt(7)*freq; \textit{%} n=-6 case, should be 4.98712e11
omega=2*pi*freq;
dx=1e-4; \textit{%} space increment of square lattice
dt=dx/(2.0*cc); \textit{%} Courant-friendly time step
\texttt{dt}=22.39e-12; \textit{%} paper time step (violates stability requirement)
nmax=5000; \textit{%} total number of time steps
\texttt{tp}=1/freq; \textit{%} Tp
m=2; \textit{%} m periods for input signal pulses
\texttt{mn}=100; \textit{%} n periods for input signal pulses
\texttt{waist}=50;
\texttt{moviesteps}=100; \textit{%} number of time steps between movie slides

\texttt{ie}=640;
\texttt{je}=830;

\texttt{is} = \texttt{ie}/2; \textit{%} location of z-directed hard source
\texttt{js}=40; \textit{%} location of z-directed hard source

A.2 Source Definitions

source=zeros(1,nmax);

gon='10*(eval(xon))^3-15*(eval(xon))^4+6*(eval(xon))^5';
goff='1-(10*(eval(xoff))^-3+15*(eval(xoff))^4+6*(eval(xoff))^5)';
xon='t/(m*tp)';
xoff='(t-(m+mn)*tp)/(m*tp)';
f1='eval(gon)*sin(omega*t)';
f2='sin(omega*t)';
f3='eval(goff)*sin(omega*t)';
A.3 Updating Coefficients For The Main Grid

In addition to the standard coefficients used in the Wagness's original FDTD2D code, we also had to include new coefficients, gamfac, ga, and gb seen here:

```matlab
for i=1:media
eaf =dt*sig(i)/(2.0*epsz*eps(i));
ca(i)=(1.0-eaf)/(1.0+eaf);
cb(i)=dt/epsz/eps(i)/dx/(1.0+eaf);
haf =dt*sim(i)/(2.0*muz*mur(i));
da(i)=(1.0-haf)/(1.0+haf);
db(i)=dt/muz/mur(i)/dx/(1.0+haf);
end
gamfac = 0.5*gamma*dt;
ga=(1-gamfac)/(1+gamfac);
gb=(dt*epsz*wpem^2)/(1+gamfac);
```

A.4 Metamaterial Slab Setup

```matlab
st = 200;  % number of cells of thickness in y-direction (height)
sbot = 240;  % number of cells from x-axis before appearance of slab
sside = 20;  % number of cells on each side of the slab (sides)
sw = ie-2*sside;  % number of cells thick in the x-direction (width)

% Ex
for i=sside+1:sw+sside
    for j=sbot:sbot+st
        caex(i,j)=ca(2);
        cbex(i,j)=cb(2);
    end
end
% Ey
for i=sside+1:sw+sside+1
    for j=sbot+1:sbot+st
        caey(i,j)=ca(2);
        cbey(i,j)=cb(2);
    end
end
```

Code used to show the slab on the graph:

```matlab
% create the box to show where the slab is on the plots
boxlinex=[sside sside+sw sside+sw sside + sside];
boxliney=[sbot sbot sbot+st sbot+st sbot];
```
A.5 Time Stepping The Source

At the beginning of each time step, the source is updated by the following code:

```matlab
t=n*dt;
if t>0 & t < m*tp
    hsource=eval(f1);
elseif t>=m*tp & t<(m+mn)*tp
    hsource=eval(f2);
elseif t>=(m+mn)*tp & t<(2*m+mn)*tp
    hsource=eval(f3);
else
    hsource=0;
end
```

A.6 Time Stepping $E_x$ Field

% vector form to make the program run more quickly
% below the slab
ex(:,2:sbot-1)=caex(:,2:sbot-1).*ex(:,2:sbot-1)+cbex(:,2:sbot-1).*(hz(:,2:sbot-1)-hz(:,1:sbot-2));
% above the slab
ex(:,sbot+st+1:je)=caex(:,sbot+st+1:je).*ex(:,sbot+st+1:je)+cbex(:,sbot+st+1:je).*(hz(:,sbot+st+1:je)-hz(:,sbot+st:je-1));
% on the left side outside the slab
% on the right side outside the slab
ex(sside+sw+1:ie,sbot:sbot+st)=caex(sside+sw+1:ie,sbot:sbot+st).*... ex(sside+sw+1:ie,sbot:sbot+st)+cbex(sside+sw+1:ie,sbot:sbot+st).*... (hz(sside+sw+1:ie,sbot:sbot+st)-hz(sside+sw+1:ie,sbot-1:sbot+st-1));
% inside the slab = add source term
ex(sside+1:sw+sside,sbot:sbot+st)=caex(sside+1:sw+sside,sbot:sbot+st).*... ex(sside+1:sw+sside,sbot:sbot+st)+cbex(sside+1:sw+sside,sbot:sbot+st).*... (hz(sside+1:sw+sside,sbot:sbot+st)-...
A.7 Time Stepping \( J_x \)

\[
J_x(sside+1:sw+sside,sbot:sbot+st) = ga*J_x(sside+1:sw+sside,sbot:sbot+st)+... \]
\[
+ gb*E_x(sside+1:sw+sside,sbot:sbot+st); \]

A.8 Time Stepping The Remaining Fields

The remaining fields are updated in the same manner that \( E_x \) and \( J_x \) were.

A.9 Time Stepping - Updating \( H_z \) With Hard Source

\[
i=(ie/2-waist:ie/2+waist)'; \]
\[
hz(ie/2-waist:ie/2+waist,js)= \]
\[
hsource*exp(-(i-0.5-ie/2).^2)/(waist^2)); \]

A.10 Visualization

Normally, a subplot would be used to show all three fields. Here, I show only the
plotting for the \( H_z \) field for bevity.

\[
timestep=int2str(n); \]
\[
pcolor(hz'); \]
\[
hold on; \]
\[
line(boxlinex,boxliney,'Color','w'); \]
\[
hold off \]
\[
shading flat; \]
\[
caxis([.1 .5]); \]
\[
axis([1 ie 1 je]); \]
\[
colorbar; \]
\[
axis image; \]
\[
axis off; \]
\[
title(['Hz at time step = ',timestep]); \]
APPENDIX B

EXCERPTS OF METAMATERIAL CLOAKING MATLAB CODE

B.1 Fundamental Constants

\[
\begin{align*}
cc &= 2.99792458 \times 10^8; \quad \text{% speed of light in free space} \\
muz &= 4.0 \times \pi \times 1.0 \times 10^{-7}; \quad \text{% permeability of free space} \\
epsz &= 1.0 / (cc \times cc \times muz); \quad \text{% permittivity of free space} \\
\text{freq} &= 2.0 \times 10^9; \quad \text{% center frequency of source excitation} \\
\lambda &= cc / \text{freq} \quad \text{% center wavelength of source excitation} \\
\text{radin} &= 100; \quad \text{% R1, inner cloak radius, 100*dx=10cm} \\
\text{radout} &= 200; \quad \text{% R2 outer cloak radius, 200*dx=20cm} \\
A &= (\text{radout} / (\text{radout} - \text{radin})) \quad \text{% used in Hz calculation} \\
\omega &= 2 \times \pi \times \text{freq}; \quad \text{% angular frequency} \\
\text{dx} &= \lambda / 150 \quad \text{% dx=dy p6725} \\
\text{dt} &= \text{dx} / (\sqrt{2} \times cc) \quad \text{% Courant-friendly time step, p6725} \\
\text{invmudtsq} &= 1 / (muz \times (\text{dt}^2)); \quad \text{% simpler term defined} \\
\text{stermsq} &= (\sin(\omega \times \text{dt} / 2))^2; \\
\text{ctermsq} &= (\cos(\omega \times \text{dt} / 2))^2; \\
\text{epsr} &= (r - \text{radin}) / r; \quad \text{% relative permittivity} \\
\text{wpsq} &= @r (-4 \times \text{stermsq} \times (\text{epsr}(r) - 1) / ((\text{dt}^2) \times \text{ctermsq})); \\
\text{wpmsq} &= @r 0; \quad \text{% magnetic plasma frequency} \\
muzr &= @r (A^2) \times ((r - \text{radin}) / r); \\
\text{wpmsq} &= @r (\omega^2) \times (1 - \text{muzr}(r) / A); \\
\text{epsphi} &= @r r / (r - \text{radin}); \quad \text{% phi permittivity depends on r} \\
\gamma &= 0.0; \quad \text{% collision frequency} \\
\text{gammapm} &= \text{gamma}; \quad \text{% magnetic collision frequency = gamma} \\
\text{excbmin} &= -100.0; \\
\text{excbmax} &= 100.0; \\
\text{eycmin} &= -20; \\
\text{eycmax} &= 20; 
\end{align*}
\]
hzcbmin = -0.2;
hzcbmax = 0.2;
nmax=2000; %total number of time steps
amp = 0.1;
media=2;
eps=[1.0 1.0];
sig=[0.0 0.0];
mur=[1.0 1.0];
sim=[0.0 0.0];

ie=800;
je=800;

% Since we are using a line source, we only need to store
% the location where it will be placed.
js=20; %location of z-directed hard source

B.2 Field Arrays

This shows the additional arrays beyond what is necessary in FDTD2D that
needed to be created.

ex=zeros(ie,jb); %fields in main grid
exprev=ex;
exprevprev=ex;
ey=zeros(ib,je);
eyprev=ey;
eyprevprev=ey;
hz=zeros(ie,je);
hzprev=hz;
hzprevprev=hz;
Dx=ex;
Dxprev=Dx;
Dxprevprev=Dx;
Dy=ey;
Dyprev=Dy;
Dyprevprev=Dy;
bz=hz;
bzprev=bz;
B.3 Updating Coefficients For The Main Grid

We are no longer using any of the original FDTD2D coefficients in this simulation.

% INSIDE CYLINDER (CLOAKING REGION) p. 6723 constants given
gambarwodt=gamma/(2*dt);
invdtsq=1/(dt^2);
ax = @(r,phi) ((sin(phi))^2)*(invdtsq+gambarwodt+(wpsq(r)/4))+... 
      epsphi(r)*((cos(phi))^2)*(invdtsq+gambarwodt);
bx = @(r,phi) ((sin(phi))^2)*(-2*invdtsq+wpsq(r)/2)-... 
      epsphi(r)*((cos(phi))^2)*2*invdtsq;
cx = @(r,phi) ((sin(phi))^2)*(invdtsq-gamovtwodt+wpsq(r)/4)+... 
      epsphi(r)*((cos(phi))^2)*2*invdtsq;
ddx = @(r,phi) (epsphi(r)*(invdtsq-gamovtwodt)-... 
      (invdtsq-gamovtwodt+wpsq(r)/4))*sin(phi)*cos(phi);
eex = @(r,phi) (epsphi(r)*(-2)*invdtsq-(-2*invdtsq+... 
      (wpsq(r)/2)))*sin(phi)*cos(phi);
fx = @(r,phi) (epsphi(r)*(invdtsq-gamovtwodt)-... 
      (invdtsq-gamovtwodt+wpsq(r)/4))*sin(phi)*cos(phi);
gx = @(r) epsz*epsphi(r)*(-2*invdtsq+(wpsq(r)/2));
hx = @(r) epsz*epsphi(r)*(invdtsq-gamovtwodt+(wpsq(r)/4));
lx = @(r) epsz*epsphi(r)*(invdtsq-gamovtwodt+(wpsq(r)/4));

ay = @(r,phi) ((cos(phi))^2)*(invdtsq+gamovtwodt+(wpsq(r)/4))+... 
      epsphi(r)*((sin(phi))^2)*(invdtsq+gamovtwodt);
by = @(r,phi) ((cos(phi))^2)*(-2*invdtsq+(wpsq(r)/2))-... 
      epsphi(r)*((sin(phi))^2)*(2*invdtsq);
cy = @(r,phi) ((cos(phi))^2)*(-2*invdtsq+(wpsq(r)/2))-... 
      epsphi(r)*((sin(phi))^2)*(2*invdtsq);
ddy = ddx;
eey = eex;
fy = fx;
gy = gx;
hy = hx;
ly = lx;
First, we need to define the center of the cylinder.

```matlab
icenter = ie/2;
jcenter = 300;
```

Then, after some vector initialization, I loop through all the nodes in the main grid to find those that are inside the cloaked, cloaking, and free space regions. The main grid is separated into three domains. We must do this for each field separately, due to using the classical Yee staggered grid.

For brevity, I will give an example for only the $E_x$ and $D_x$ fields.

```matlab
tempx = i+0.5-icenter;  % original fdtd2d code
tempy = j-jcenter;      % original fdtd2d code
dist2 = tempx^2 + tempy^2;
if dist2 < radout^2 & dist2 > radin^2
    exinc = exinc + 1;
exin1(exinc,:)=\[i j\];
    if tempy==0 & tempx>0
        phiex = 0;
    elseif tempy==0 & tempx<0
        phiex = pi;
    elseif tempx==0 & tempy>0
        phiex = pi/2;
    elseif tempx==0 & tempy<0
        phiex = 3*pi/2;
    elseif tempx>0 & tempy>0 \%Q1
        phiex = atan(tempy/tempx);
    elseif tempx<0 & tempy>0 \%Q2
        phiex = pi-abs(atan(tempy/tempx));
    elseif tempx<0 & tempy<0 \%Q3
        phiex = pi+abs(atan(tempy/tempx));
    else \%Q4
        phiex = 2*pi-abs(atan(tempy/tempx));
end
radex = sqrt(dist2);
coefax(exinc)=ax(radex,phiex);
coefbx(exinc)=bx(radex,phiex);
coefcx(exinc)=cx(radex,phiex);
coefddx(exinc)=ddx(radex,phiex);
```
coefeex(exinc)=eex(radex,phiex);
coeffx(exinc)=fx(radex,phiex);
coefgx(exinc)=gx(radex);
coefhx(exinc)=hx(radex);
coeflx(exinc)=lx(radex);
elseif dist2 >= radout^2 % freespace
  exoutc = exoutc + 1;
  exout1(exoutc,:) = [i j];
else % it’s in the PEC area
  end

You can see that we are storing the radius and angle (cylindrical coordinates) for each node that lies inside the cloaking region. If the node lies inside the inner, cloaked, region, we do nothing. Those nodes will not be updated at all, as they should remain null value throughout the calculations. The location of the nodes, radius, and angle are calculated similarly for the remaining fields.

The nodes that are found to be inside the cloaking region and in the free space region were stored in temporary arrays. After their sizes are known, they are stored in permanent arrays:

- exincl - Nodes which will be used for $E_x$ calculation that are inside the cloaking region
- eyincl - Nodes which will be used for $E_y$ calculation that are inside the cloaking region
- hzincl - Nodes which will be used for $H_z$ calculation that are inside the cloaking region
- exout - Nodes which will be used for $E_x$ calculation that are in free space.
- eyout - Nodes which will be used for $E_y$ calculation that are in free space.
• hzout - Nodes which will be used for $H_z$ calculation that are in free space.

B.5 Time Stepping - Storing Previous Values

Since our algorithm requires storage of the previous values for all fields used, we begin the time loop by updating those values.

```matlab
Dxprevprev = Dxprev;
Dxprev = Dx;
Dyprevprev = Dyprev;
Dyprev = Dy;
exprevprev = exprev;
exprev = ex;
eyprevprev = eyprev;
eyprev = ey;
dbarxprevprev = dbarxprev;
dbarxprev = dbarx;
dbaryprevprev = dbaryprev;
dbaryprev = dbary;
bzprevprev = bzprev;
bzprev = bz;
hzprevprev = hzprev;
hzprev = hz;
```

B.6 Time Stepping Electric Flux Densities ($D_x$ and $D_y$)

```matlab
Dx(:,2:je) = Dxprev(:,2:je)+(dt/dx)*(hz(:,2:je)-hz(:,1:je-1));
Dy(2:ie,:) = Dyprev(2:ie,:)-(dt/dx)*(hz(2:ie,:)-hz(1:ie-1,:));
```

B.7 Time Stepping $E_x$ Field

```matlab
% Ex
% outside cloaking region
for i=1:exoutc
    ii = exout(i,1);
    jj = exout(i,2);
    ex(ii,jj) = (1/(epsz*invdtsq))*(invdtsq*Dx(ii,jj)+... -2*invdtsq*Dxprev(ii,jj)+... invdtsq*Dxprevprev(ii,jj)-... (-2*epsz*invdtsq*exprev(ii,jj)+... epsz*invdtsq*exprevprev(ii,jj)));
end
```
for i=1:exinc
    ii=exincl(i,1);
    jj=exincl(i,2);
    % if jj ~ jb
    dbary(ii,jj) = 0.25*(Dy(ii,jj-1)+Dy(ii+1,jj-1)+Dy(ii,jj)+Dy(ii+1,jj));
    ex(ii,jj)=(1/coeflx(i))*(coefax(i)*Dx(ii,jj)+coefbx(i)*Dxprev(ii,jj)+coefcx(i)*Dxprevprev(ii,jj)+coefddx(i)*dbary(ii,jj)+coefeex(i)*dbaryprev(ii,jj)+coeffx(i)*dbaryprevprev(ii,jj)-(coefgx(i)*exprev(ii,jj)+coefhx(i)*exprevprev(ii,jj)));
% end
end

Time stepping for the $E_y$ field is done in the same way as that of $E_x$, so I omit for brevity.

B.8 Time Stepping Magnetic Flux Density $B_z$

% apply periodic boundary conditions on bz
bz(2:ie-1,1:je)=bzprev(2:ie-1,1:je)+(dt/dx)*(ex(2:ie-1,2:jb)-ex(2:ie-1,1:je)+ey(2:ie-1,1:je)-ey(3:ie,1:je));
bz(1,:) = bz(ie-1,:);
bz(ie,:) = bz(2,:);

B.9 Time Stepping $H_z$ Field

% outside CLOAKING region in FREESPACE
for i=1:hzoutc
    ii=hzout(i,1);
    jj=hzout(i,2);
    hz(ii,jj)=(invmudtsq*bz(ii,jj)-2*invmudtsq*bzprev(ii,jj)+invmudtsq*bzprevprev(ii,jj)+2*invdtsq*hzprev(ii,jj)-invdtsq*hzprevprev(ii,jj))/invdtsq;
end

% inside CLOAKING region
for i=1:hzinc
ii=hzincl(i,1);
jj=hzincl(i,2);
r=radhz(i);

hz(ii,jj)=(1/A)*((invmudtsq+(gamma/(2*muz*dt)))*bz(ii,jj)-... 
  2*invmudtsq*bzprev(ii,jj)+... 
  (invmudtsq-(gamma/(2*muz*dt)))*bzprevprev(ii,jj)+... 
  A*(2*invdtsq-(0.5*wpmsq(r)))*hzprev(ii,jj)-... 
  A*(invdtsq-gamovtwodt+0.25*wpmsq(r))*hzprevprev(ii,jj)/... 
  (invdtsq+gamovtwodt+0.25*wpmsq(r));
end

B.10 Time Stepping - Updating $H_z$ With Hard Source

$hz(1:ie,js)=amp*sin(omega*t)*ones(ie,1);$

B.11 Visualization

The visualization used in this code is the same as is used in the metamaterial code (see Appendix 1).
C.1 Transformation Optics

Here is the derivation of the transformed material parameters in [49] in which I utilized the more helpful explanations that were given in a later paper by some of the same authors [51].

We start with the basic spherical coordinate transformation equations:

\[
\begin{align*}
\quad x &= r \sin \theta \cos \phi \\
\quad y &= r \cos \theta \sin \phi \\
\quad z &= r \cos \theta .
\end{align*}
\]

The partial derivatives are as follows.

\[
\begin{align*}
\quad \frac{\partial x}{\partial r} &= \sin \theta \cos \phi & \quad \frac{\partial x}{\partial \theta} &= r \cos \theta \cos \phi & \quad \frac{\partial x}{\partial \phi} &= -r \sin \theta \sin \phi \\
\quad \frac{\partial y}{\partial r} &= \sin \theta \sin \phi & \quad \frac{\partial y}{\partial \theta} &= r \cos \theta \sin \phi & \quad \frac{\partial y}{\partial \phi} &= r \sin \theta \cos \phi \\
\quad \frac{\partial z}{\partial r} &= \cos \theta & \quad \frac{\partial z}{\partial \theta} &= -r \sin \theta & \quad \frac{\partial z}{\partial \phi} &= 0
\end{align*}
\]

Now,

\[
\begin{align*}
Q_r^2 &= \left( \frac{\partial x}{\partial r} \right)^2 + \left( \frac{\partial y}{\partial r} \right)^2 + \left( \frac{\partial z}{\partial r} \right)^2 = \sin^2 \theta \cos^2 \phi + \sin^2 \theta \sin^2 \phi + \cos^2 \theta = 1 \quad (C.1.5) \\
Q_\theta^2 &= \left( \frac{\partial x}{\partial \theta} \right)^2 + \left( \frac{\partial y}{\partial \theta} \right)^2 + \left( \frac{\partial z}{\partial \theta} \right)^2 = r^2 \quad (C.1.6) \\
Q_\phi^2 &= \left( \frac{\partial x}{\partial \phi} \right)^2 + \left( \frac{\partial y}{\partial \phi} \right)^2 + \left( \frac{\partial z}{\partial \phi} \right)^2 = r^2 \sin^2 \theta . \quad (C.1.7)
\end{align*}
\]
So, by \[49\text{ Equation}(2)\],

\[\epsilon_s = \frac{(1)(r)(r \sin \theta)}{l^2} = \epsilon r^2 \sin \theta \quad (C.1.8)\]

\[\epsilon_s = \epsilon \sin \theta \quad (C.1.9)\]

\[\epsilon_s = \frac{\epsilon}{\sin \theta} \quad (C.1.10)\]

which can be written as

\[\epsilon_s = \epsilon \begin{pmatrix} r^2 \sin \theta & 0 & 0 \\ 0 & \sin \theta & 0 \\ 0 & 0 & \frac{1}{\sin \theta} \end{pmatrix} \quad (C.1.11)\]

This is in similar form to \[51\text{ Equation } (31)\]. The values in these entries are those that we will use later to renormalize the final material parameters.

Note the linear transformation, \(r' = R_1 + r \left( \frac{R_2 - R_1}{R_2} \right)\). Solving for \(r\) gives

\(r = (r' - R_1) \left( \frac{R_2}{R_2 - R_1} \right), \text{ and } \frac{\partial r}{\partial r'} = \frac{R_2}{R_2 - R_1}, \) then

\[\epsilon'_{s_r} = \frac{R_2}{R_2 - R_1} (\frac{(1)(1)}{2}) \epsilon r^2 \sin \theta = \frac{R_2 - R_1}{R_2} \epsilon r^2 \sin \theta \quad (C.1.12)\]

\[\epsilon'_{s_\theta} = \frac{R_2}{R_2 - R_1} \epsilon \sin \theta \quad (C.1.13)\]

\[\epsilon'_{s_\phi} = \frac{R_2}{R_2 - R_1} \frac{\epsilon}{\sin \theta} \quad (C.1.14)\]

Substitute the definitions of \(r = (r' - R_1) \left( \frac{R_2}{R_2 - R_1} \right), \theta = \theta', \) and \(\phi = \phi'\) into Equations \(C.1.12\) \(C.1.14\) to get

\[\epsilon'_{s_r} = \left( \frac{R_2}{R_2 - R_1} \right) (r' - R_1)^2 \sin \theta' \epsilon \quad (C.1.15)\]

\[\epsilon'_{s_\theta} = \frac{R_2}{R_2 - R_1} \sin \theta' \epsilon \quad (C.1.16)\]

\[\epsilon'_{s_\phi} = \frac{R_2}{R_2 - R_1} \frac{\epsilon}{\sin \theta'} \quad (C.1.17)\]
Now, we renormalize:

\[
\begin{align*}
\epsilon_{r'} &= \frac{\epsilon_{r'}}{(r')^2 \sin \theta' \epsilon} = \left( \frac{R_2}{R_2 - R_1} \right) \frac{(r' - R_1)^2}{(r')^2}, \\
\epsilon_{\theta'} &= \frac{\epsilon_{\theta'}}{\sin \theta' \epsilon} = \frac{R_2}{R_2 - R_1}, \\
\epsilon_{\phi'} &= \frac{\epsilon_{\phi'}}{\frac{\epsilon}{\sin \theta}} = \frac{R_2}{R_2 - R_1},
\end{align*}
\]

(C.1.18) (C.1.19) (C.1.20)

which you can see shows a typo in [19, Equation (7)] in the equation for \( \epsilon_{r'} \).
BIBLIOGRAPHY


VITA

Graduate College
University of Nevada, Las Vegas

Valjean Elizabeth Elander

Degrees:
Bachelor of Science in Mathematics, 1999
University of Missouri, St. Louis

Master of Science in Mathematics, 2005
University of Nevada, Las Vegas

Special Honors and Awards:
$10,000 NASA EPSCoR Student Travel Grant, 2004

Publications:


A Preliminary ZEUS Lightning Location Error Analysis Using a Modified Retrieval Theory, 2004 Fall Meeting of the American Geophysical Union, San Francisco, California, December 13-17, 2004, Presenter


Dissertation Title:
Mathematical Modeling of Metamaterials

Dissertation Examination Committee:
Chairperson, Jichun Li, Ph.D.
Committee Member, Michael Marcozzi, Ph.D.
Committee Member, Monika Neda, Ph.D.
Committee Member, Pengtao Sun, Ph.D.
Committee Member, Hongtao Yang, Ph.D.
Graduate Faculty Representative, Yitung Chen, Ph.D.