FINITE-DIFFERENCE TIME-DOMAIN SIMULATIONS OF METAMATERIALS

A Dissertation Presented
by
Zhengwei Hao
to
The Electrical and Computer Engineering Department

in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

In the field of

Electrical Engineering

Northeastern University
Boston, Massachusetts
December, 2012
Abstract

Metamaterials are periodic structures created by many identical scattering objects which are stationary and small compared to the wavelength of electromagnetic wave applied to it so that when combined with different elements, these materials have the potential to be coupled to the applied electromagnetic wave without modifying the structure. Due to their unusual properties that are not readily available in nature, metamaterials have been drawing significant attentions in many research areas, including theoretical, experimental as well as numerical investigations.

As one of the major computational electromagnetic modeling methods, finite-difference time-domain (FDTD) technique tackles problems by providing a full wave solution. FDTD, which is able to show transient evolution of interactions between electromagnetic wave and physical objects, not only has the advantage in dispersive and nonlinear material simulations, but also has the ability to model circuit elements including semiconductor devices. All these features make FDTD a competitive candidate in numerical methods of metamaterial simulations.

This dissertation presents the implementation of FDTD technique to deal with three dimensional (3D) problems characterized with metamaterial structures. We endeavor to make the FDTD engine multi-functional and fast, as depicted in the following three efforts:
(1) We incorporated FDTD engine with the stable and highly efficient model for materials with dispersion, nonlinearity and gain properties.

(2) We coupled FDTD engine with SPICE, the general-purpose and powerful analog electronic circuit simulator. This makes FDTD ready to simulate complex semiconductor devices and provides a variety of possibilities for novel metamaterials.

(3) We investigated the cutting-edge area of Graphics Processing Units (GPU) computing module to speed up the FDTD engine, and implemented subgridding system to target more efficient modeling for metamaterial applications with embedded fine structures.

The contribution of this work is toward the development of a powerful FDTD engine for modern metamaterial analysis. Our implementation could be used to improve the analysis of a number of electromagnetic problems.
## Contents

Abstract ........................................................................................................................................... i

Contents ........................................................................................................................................... iii

List of Figures .................................................................................................................................. vi

AKNOWLEDGEMENTS ................................................................................................................ ix

Chapter 1 ......................................................................................................................................... 1

  1.1 Electromagnetic Metamaterials ............................................................................................... 1
     1.1.1 Applications of EM Metamaterial .................................................................................. 3
     1.1.2 Analysis in Metamaterial Application ............................................................................. 5
  1.2 FDTD Method Basics ............................................................................................................. 6
     1.2.1 Yee Algorithm ............................................................................................................... 7
     1.2.2 Absorbing Boundary Condition ..................................................................................... 11
     1.2.3 Total Field and Scattered Field Model .......................................................................... 12
     1.2.4 Periodic Structure ......................................................................................................... 13
     1.2.5 FDTD Algorithm Acceleration ..................................................................................... 13
  1.3 Topics and Chapters to Cover ................................................................................................. 14

Chapter 2 ......................................................................................................................................... 16

  2.1 ADE Method for Linear Dispersive Material .......................................................................... 17
     2.1.2 Material with Multiple Debye Poles .............................................................................. 18
     2.1.3 Material with Multiple Lorentz Poles ........................................................................... 21
     2.1.4 Material with Multiple Drude Poles ............................................................................. 24
  2.2 Implementation and Verifications ........................................................................................... 27
  2.3 Application and Numerical Results ........................................................................................ 31

Chapter 3 ......................................................................................................................................... 36

  3.1 Nonlinear Dispersive Material ............................................................................................... 36
  3.2 Algorithm for Nonlinear Dispersive Material ......................................................................... 36
     3.2.1 Contribution of Linear Debye Polarization ................................................................. 37
6.2 Cartesian Subgrids........................................................................................................... 89
  6.2.1 Introduction to the Geometry ................................................................................. 90
  6.2.2 Implementation Scheme ....................................................................................... 92
  6.3 Numerical Results ....................................................................................................... 95
Chapter 7 ......................................................................................................................... 98
Conclusion ......................................................................................................................... 98
  7.1 Summary .................................................................................................................. 98
  7.2 Future Research ....................................................................................................... 99
Bibliography ..................................................................................................................... 101
List of Figures

Figure 1.1 FDTD grid unit, Yee cell illustration .......................................................... 8
Figure 1.2 General FDTD field update procedure ......................................................... 10
Figure 1.3 Illustration of CPML effect on EM wave reflection ................................. 11
Figure 1.4 (A) Total field/scattered field technique effects on EM wave scattering (B) Matlab simulation for a square PEC object .......................... 12

Figure 2.1 FDTD field update procedure for dispersive material ......................... 18
Figure 2.2 FDTD field update procedure for Debye material ................................. 20
Figure 2.3 FDTD field update procedure for Lorentz material .............................. 24
Figure 2.4 FDTD field update procedure for Drude material ................................. 27
Figure 2.5 FDTD field update procedure for dispersive materials with PML calculation ........................................................... 28
Figure 2.6 Drude material slab with electric negative property .......................... 28
Figure 2.7 Magnitude of transmission coefficient for the ENG medium .......... 29
Figure 2.8 Phase of transmission coefficient for the ENG medium ..................... 30
Figure 2.9 Magnitude of reflection coefficient for Lorentz material ............ 31
Figure 2.10 Layered substrates with plasmonic array nano-antennas on top.. 32
Figure 2.11 Nano-shell particles with dielectric as core and silver as shell.. 33
Figure 2.12 Normalized radiated power obtained (A) in XY plane, and (B) in YZ plane for nano-shells depicted in Fig.2.11 .............................................................. 34
Figure 2.13 Magnitude of EZ(dB) for the nanoantenna in Fig.2.11 in XY (Z=0) plane and YZ (X=0) plane. (A) Theory, (B) FDTD ........................................... 35

Figure 3.1 FDTD field update procedure for nonlinear material ....................... 43
FIGURE 3.2 GAUSSIAN WAVE PACK AFTER PROPAGATING 55UM AND 126UM IN LINEAR LORENTZ MATERIAL ................................................................. 45

FIGURE 3.3 GAUSSIAN OPTICAL WAVE PACK AFTER PROPAGATING 55UM AND 126UM IN NONLINEAR DISPERSIVE MATERIAL ................................................................. 45

FIGURE 3.4 FDTD FIELD UPDATE PROCEDURE FOR MACROSCOPIC OPTICAL GAIN MATERIAL ............................................................................................................. 49

FIGURE 3.5 COMPARISON FOR AMPLIFICATION FACTOR OF OPTICAL GAIN MATERIAL ........ 50
FIGURE 3.6 COMPARISON FOR PHASE FACTOR OF OPTICAL GAIN MATERIAL.................. 50

FIGURE 4.1 UNIT FDTD CELL ILLUSTRATION FOR LUMPED CIRCUIT CURRENT CALCULATION ................................................................................................................. 53

FIGURE 4.2 FDTD FIELD UPDATE PROCEDURE FOR CIRCUIT ELEMENTS ..................... 54
FIGURE 4.3 FDTD CIRCUIT SIMULATION WITH RESISTIVE VOLTAGE SOURCE AND DIODE... 57
FIGURE 4.4 VOLTAGE OUTPUT FOR DIODE ELEMENT .................................................. 57
FIGURE 4.5 CIRCUIT MODEL FOR AMPERE’S LAW EQUATION THAT CONNECTS FDTD AND SPICE ............................................................................................................. 61

FIGURE 4.6 FDTD FIELD UPDATE PROCEDURE FOR COUPLING WITH SPICE ............... 62
FIGURE 4.7 FLOWCHART FOR FDTD-SPICE SIMULATION ........................................... 63
FIGURE 4.8 RLC CIRCUIT CONFIGURATION .................................................................. 65
FIGURE 4.9 RLC CIRCUIT IMPEDANCE Z BEHAVIOR. .................................................... 65
FIGURE 4.10 TRANSISTOR CIRCUIT CONFIGURATION...................................................... 66
FIGURE 4.11 OUTPUT VOLTAGE WAVEFORM IN TIME. THE FDTD-SPICE SIMULATION MATCHED VERY WELL WITH PURE SPICE SIMULATION ........................................... 67

FIGURE 4.12 METAMATERIAL CONSTRUCTED FROM ARRAY OF LOOPS TERMINATED TO CAPACITORS Cp ................................................................. 68
FIGURE 4.13 TRANSMISSION COEFFICIENT OF METAMATERIAL CONSTRUCTED FROM ARRAY OF LOOPS TERMINATED TO CAPACITORS Cp ................................................. 69
Figure 4.14 Basic non-Foster element: NIC configuration based on Linvill’s design, and equivalent circuit model for the loop-based metamaterial integrated with NIC negative inductor ................................................................. 71

Figure 4.15 Transmission coefficient for metamaterial of loops terminated with I: \( C_P=0.13\, pF, \, L=4\, nH \); II: \( C_P=0.2\, pF, \, L=4\, nH \), NIC circuit \( L_N=-2\, nH \); III: \( C_P=0.2\, pF, \, L=2\, nH \) ........................................................................................................................................ 72

Figure 4.16 Metasurface structure with periodic square metal patches and its unit cell that loaded with varactor diodes ................................................................. 74

Figure 4.17 Reflection phase for the metasurface. Controlling the bias of the diode tunes the resonance frequency ........................................................................... 75

Figure 4.18 Performance data for SMV1232 Diode Varactor ......................... 76

Figure 5.1 Acceleration by parallel computation ............................................. 80

Figure 5.2 FDTD calculation procedure for OpenCL implementation .......... 83

Figure 5.3 Speedup of OpenCL implementation for FDTD simulation compared to CPU version ........................................................................................................ 84

Figure 6.1 Two dimensional view of 2:1 subgridding system ......................... 90

Figure 6.2 Three dimensional view of 2:1 subgridding system .................... 91

Figure 6.3 Update procedure for 2:1 subgridding system ............................ 93

Figure 6.4 Half space Lorentz material embedded with 2:1 subgridding system ......................................................................................................................... 96

Figure 6.5 Reflection of half space Lorentz material calculated with 2:1 subgridding system ....................................................................................................... 97
I would like to express my sincere gratitude and respect to the many people who have made this work possible and helped me a lot during my doctoral program.

I extend my deepest gratitude to my advisor, Dr. Hossein Mosallaei. I’m grateful for the thorough guidance he provided through my doctoral study and research. His vast knowledge and research experience impacted, inspired and helped me a lot. This work would not have been possible without his great support and encouragement.

I would like to thank Dr. David R. Kaeli, Dr. Hossein Mosallaei and Dr. Philip E. Serafim for being my committee members. I would like to express my thanks to Dr. David R. Kaeli for being very supportive on my dissertation writing. I would also like to thank my teacher Dr. Philip E. Serafim, who provided excellent teaching on electromagnetics.

I express my sincere gratitude to the group members at our office who provided me a pleasant environment for my study and research.

I’m grateful for ECE staff Ms. Faith Crisley, Ms. Sharon Heath and Mr. Fernando Dellaporta, for their kind assistance during my study at Northeastern University.

Finally, I’d like to dedicate this work to my parents, Shigeng Hao and Chuanlan Zhang, who brought me up with endless love and taught me to be a better person.
Chapter 1

Introduction

1.1 Electromagnetic Metamaterials

In 1968, Victor Veselago theoretically analyzed the propagation of plane wave in a material with both negative permittivity and negative permeability in his original research, which is the first study in this area and key to the advancement of recent researches in electromagnetic (EM) metamaterials [1].

In the literatures [2, 3, 4], electromagnetic metamaterials are defined as artificial effectively homogeneous electromagnetic structures with unusual properties not readily available in nature. Metamaterial gain its properties from its structure rather than directly from its composition. Typical metamaterials are periodic structures created by many identical resonant scattering objects which are stationary and small compared with the wavelength of electromagnetic radiation. For conventional materials in nature, the basic unit is the constitutive molecules, which are of the order of the angstrom (0.1nm) [2]. In periodic metamaterials, the unit cell size $p$ is of the order of centimeter. Thus in L-X microwave band ranges, the electric size of the structure unit is $p/\lambda_s \approx 10^{-9}$ for natural dielectric and about $10^{-2}$ for metamaterial. This number is small enough for the refraction
of the electromagnetic wave interacting with metamaterials. This feature makes metamaterial act like a real homogeneous material, which can be characterized by the effective macroscopic parameters $\varepsilon$ and $\mu$.

When electromagnetic waves interact with the inclusions of a metamaterial, electric and magnetic moments are induced. These induced moments will in turn affect the effective parameters $\varepsilon$ and $\mu$. Since almost all the natural materials have positive values for both $\varepsilon$ and $\mu$, the greatest potential of metamaterials lies in the possibility to compose structure with effective parameters: negative $\varepsilon$ and negative $\mu$. This double negative (DNG) metamaterial has a negative refractive index $n = -\sqrt{\varepsilon \mu}$ and displays unusual refractive properties. For DNG metamaterials, Snell’s law ($n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$) still apply. But because $n_2$ is negative, the rays will be refracted on the same side of the normal when entering the material. Moreover, for plane waves travelling in this metamaterials, the electric field $E$, magnetic field $H$ and wave vector $k$ will follow a left-hand rule, which is different from the right-hand rule for conventional materials [4, 5].

For actual material, the permittivity and permeability have complex values and don’t need to have a negative real part to display negative refraction. Based on the different values of interest, electromagnetic metamaterials have the following several primary classes. If both permittivity and permeability of a media are negative, then it is double negative metamaterial or negative-index metamaterial (NIM). When only one of the permittivity and permeability is negative, it’s called single negative (SNG) metamaterial. SNG can be further classified into two kinds: epsilon negative (ENG) material which has negative permittivity and mu negative (MNG) material that have
negative permeability. There are also double positive (DPS) material, electromagnetic bandgap (EBG) material which are usually achieved by photonic crystals (PC) and left-handed (LHM) material, Chiral material, bi-isotropic and bi-anisotropic material.

1.1.1 Applications of EM Metamaterial

There are many ways to generate different effective permittivity and permeability for a metamaterial. For example, changing host materials; varying the size, shape, and composition of the inclusions; changing the density, arrangement, and alignment of these inclusions [3]. These degrees of freedom make the metamaterial tunable in its properties and thus provide us a variety of applications for metamaterial.

For example, a tunable metamaterial may have the capability to adjust frequency changes in the refractive index at will, providing a possibility to expand beyond the bandwidth limitations in left-handed materials. Plasmonic metamaterials are negative index metamaterials and can be used to couple the incident light with the metal dielectric material to create self-sustaining, propagating electromagnetic waves known as surface plasmon polaritons [6].

There are also metamaterial antennas, which are commercially available already. In this application, metamaterials are used to improve the performance of the antenna systems by providing properties such as electrically small antenna size, high directivity, and tunable operational frequency. Metamaterials can be integrated with nonlinear media to provide the option to change the power of the incident wave, which overcomes the limitation of many common optical materials that have weak nonlinearity.
Another interesting and potentially promising application is metamaterial circuit. As we know, the power and thermal limits of complementary metal-oxide semiconductor (CMOS) technology have been an important bottleneck for the microprocessor vendors to scale the frequency of the electronic devices to derive more performance [7]. The limit of the circuit speed itself, in electronic devices such as computers and cellphones, however, is the RC time constant \( \tau = RC = \frac{1}{2\pi f_c} \), where \( f_c \) is the cutoff frequency, while R and C are the resistance and capacitance, respectively. To make the circuit work faster, the RC time constant must be smaller. Smaller scale objects, such as metamaterials, provide a possibility to this. For example, if we want to design a computer that has clock frequency as 300GHz, there’s big chance that our normal resistors and capacitors won’t work well on this machine. It’s common to have a capacitor valued at picofarads, but a one femtofarad capacitor would be much harder to achieve. Conventional RLC circuits usually works at frequencies no more than 50GHz. Compared to these circuits operating at low frequency, terahertz metamaterials are designed to work at frequencies range from 0.1THz to 10 THz, which shows a promise for faster communication.

Besides that, a problem that may be even more important is the power consumed by the computer. People have to design proper power dissipation system for the CPU. This is another important issue related to the design, which is an advantage of metamaterials that may generate less power waste.
1.1.2 Analysis in Metamaterial Application

Based on the definition and application of metamaterial, one fundamental method to analyze metamaterial characteristics relies on the material parameters: the effective permittivity \( \varepsilon_r \) and the effective permeability \( \mu_r \). Materials with negative parameters are innately dispersive and their permittivity and permeability will alter with respect to frequency [5]. At the meanwhile, many new applications also use nonlinear and gain materials to tune the properties of the metamaterials. Therefore, models that deal with dispersion, nonlinearity and gain property play an important role in the analysis of metamaterial application.

Transmission line (TL) theory, which uses discrete lumped-circuit model to explain characteristic impedance and other electromagnetic properties of a media, is another popular way to investigate metamaterials [2, 3]. The nature of TL theory dealing with lumped circuits makes metamaterials readily to incorporate many other complex elements, passive and active, linear and nonlinear, to potentially build materials that can be dynamically tuned at will.

Analysis for metamaterial applications can be done either theoretically, experimentally or numerically. Theoretical research is the foundation for the other two and provides theory support and guidance for them. But it is not always effective, especially when tackling electromagnetic problems with complex structures. While experimental work is the only reliable way to verify theories and numerical results, it usually takes much longer time and consumes more resources than the other two. Finally, along with the progressive development of computer technology, computational
electromagnetics have extensively accelerated the research process in all the aspects of modern electromagnetics. Numerical methods such as finite element method (FEM), method of moments (MoM) and finite-difference time-domain (FDTD) methods have been very successfully applied to model electromagnetic problems.

In this dissertation, FDTD method will be used to model structures characterized with electromagnetic metamaterials. Several modules for dispersive materials and SPICE interface will be implemented to incorporate the new features of metamaterial while others, such as, GPU calculation and subgrid implementations, will be developed to accelerate the simulation process.

1.2 FDTD Method Basics

As one major computational electromagnetic modeling method, Finite-difference time-domain technique tackles problems by providing a full wave solution. Since it naturally treats dispersive and nonlinear material, and covers a wide range frequencies in one single run, FDTD has been a primary means to numerically model scientific and engineering problems, like antennas, photonic crystals, transmission lines, circuit simulations and so on [8].

Maxwell’s equations, published in James Clerk Maxwell’s great work in 1861, which consist of Gauss’s Law, Gauss’s Law for magnetism, Faraday equation and Ampere’s Law, contribute to form the foundation of classical electrodynamics. Thus all the classical electromagnetics as well as the computational electromagnetic methods are based on this set of equations.
Equation (1.1) and (1.2) are the differential form of Ampere’s Law and Faraday equations, respectively. These two equations are used to solve numerical calculation in FDTD method.

\[
\nabla \times H = \varepsilon \frac{\partial E}{\partial t} + \sigma E \tag{1.1}
\]

\[
\nabla \times E = -\mu \frac{\partial H}{\partial t} \tag{1.2}
\]

where \( E \) and \( H \) are the electric filed and magnetic field, respectively; \( \varepsilon \) and \( \mu \) are the electrical permittivity and magnetic permeability, respectively; \( \sigma \) and is the electric conductivity loss; \( t \) is the time.

1.2.1 Yee Algorithm

The well-known Yee algorithm comes from Kane Yee’s publication in 1966 [9], which described the basis of the modern FDTD model to solve Maxwell’s curl equations in time domain. The special grid cell that Yee originally used is called Yee cell and it discretizes the three dimensional space into cubes.

Fig.1.1 shows the unit cell of Yee lattice. As can be seen from the figure, electric fields are along the edges of the cube while the magnetic fields are located at the center of the cube’s 6 surfaces. Each unit cell has 12 electric fields and 6 magnetic fields. Since each electric field is shared by the adjacent 4 unit cells, each unit cell possesses 3 electric fields, namely \( Ex, Ey \) and \( Ez \). Similar rule applies to magnetic fields and each unit cell possesses 3 magnetic fields, \( Hx, Hy \) and \( Hz \). In a 3D point of view, each electric field \( E \) is
surrounded by 4 circulating $H$ fields, which interprets the Ampere’s Law. Similarly, every magnetic field $H$ is surrounded by 4 circulating $E$ fields, representing Faraday equation.

![Figure 1.1 FDTD grid unit, Yee cell illustration](image)

In three-dimensional case, each of these two vector equations Eq.(1.1) and Eq.(1.2) can be converted into 3 scalar equations, forming 6 scalar equations in total:

\[
\begin{align*}
\frac{\partial E_x}{\partial t} &= \frac{1}{\varepsilon} \left( \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - \sigma E_x \right) \\
\frac{\partial E_y}{\partial t} &= \frac{1}{\varepsilon} \left( \frac{\partial H_z}{\partial z} - \frac{\partial H_x}{\partial x} - \sigma E_y \right) \\
\frac{\partial E_z}{\partial t} &= \frac{1}{\varepsilon} \left( \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - \sigma E_z \right) \\
\frac{\partial H_x}{\partial t} &= -\frac{1}{\mu} \left( \frac{\partial E_y}{\partial y} - \frac{\partial E_z}{\partial z} + \sigma_m H_x \right) \\
\frac{\partial H_y}{\partial t} &= -\frac{1}{\mu} \left( \frac{\partial E_z}{\partial z} - \frac{\partial E_x}{\partial x} + \sigma_m H_y \right) \\
\frac{\partial H_z}{\partial t} &= -\frac{1}{\mu} \left( \frac{\partial E_x}{\partial x} - \frac{\partial E_y}{\partial y} + \sigma_m H_z \right)
\end{align*}
\] (1.3) (1.4)
For a function $f$ of space and time, such as

$$f(x, y, z, t) = f(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = f^n(i, j, k)$$

(1.5)

where $\Delta t$ is the time increment, $\Delta x, \Delta y, \Delta z$ are the space increments; $n$ is the time mark of current calculation, $i, j$ and $k$ denote the space location of current calculation, the differential form in time and space can be written as

$$\frac{\partial f(x, y, z, t)}{\partial t} \bigg|_{t=n\Delta t} \approx \frac{f^{n+\frac{1}{2}}(i, j, k) - f^{n-\frac{1}{2}}(i, j, k)}{\Delta t}$$

(1.6)

$$\frac{\partial f(x, y, z, t)}{\partial x} \bigg|_{x=i\Delta x} \approx \frac{f^n(i + \frac{1}{2}, j, k) - f^n(i - \frac{1}{2}, j, k)}{\Delta x}$$

(1.7)

To make the time domain update for the above Maxwell’s equations of differential form, similar manipulations are applied. Those scalar equations from Eq.(1.3) and Eq.(1.4), taking $E_x$ and $H_x$ for example, can be written as

$$E_x^{n+\frac{1}{2}}(i, j, k) = CA(m) \cdot E_x^n(i - \frac{1}{2}, j, k)$$

$$+ CB(m) \cdot \left\{ \frac{H_z^{n+\frac{1}{2}}(i, j + \frac{1}{2}, k) - H_z^{n+\frac{1}{2}}(i, j - \frac{1}{2}, k)}{\Delta y} \right.\right.$$

$$- \left.\frac{H_y^{n+\frac{1}{2}}(i, j, k + \frac{1}{2}) - H_y^{n+\frac{1}{2}}(i, j, k - \frac{1}{2})}{\Delta z} \right\}$$

$$E_y^{n+\frac{1}{2}}(i, j, k) = \ldots..$$

$$E_z^{n+\frac{1}{2}}(i, j, k - \frac{1}{2}) = \ldots..$$

(1.8)
\[ H_x^{n+\frac{1}{2}}(i, j-\frac{1}{2}, k-\frac{1}{2}) = CC(m) \cdot H_x^{n-\frac{1}{2}}(i, j-\frac{1}{2}, k-\frac{1}{2}) + CD(m) \cdot \left( \frac{E_z^n(i, j+\frac{1}{2}, k-\frac{1}{2}) - E_z^n(i, j-\frac{1}{2}, k-\frac{1}{2})}{\Delta y} \right) \]

\[ -\frac{E_z^n(i, j-\frac{1}{2}, k+\frac{1}{2}) - E_z^n(i, j-\frac{1}{2}, k-\frac{1}{2})}{\Delta z} \]

\[ H_y^{n+\frac{1}{2}}(i-\frac{1}{2}, j, k-\frac{1}{2}) = \ldots \]

\[ H_z^{n+\frac{1}{2}}(i-\frac{1}{2}, j-\frac{1}{2}, k) = \ldots \]

where \( CA, CB, CC \) and \( CD \) are coefficients, \( m \) is the number of material.

After discretization of the space, as well as time derivatives, scalar form of Eq.(1.8) and Eq.(1.9) are used for the electric and magnetic fields update at different time steps, respectively. Thus, one by another, the electric and magnetic fields are calculated as time goes on, as shown in Fig.1.2.

---

**Figure 1.2 General FDTD field update procedure**
1.2.2 Absorbing Boundary Condition

In the numerical calculations, all the data, including cell discretization, electric/magnetic fields, material parameters, are stored in the computer memory. However, any computer has a memory limit. As a result, our FDTD model must have a boundary. Usually, this outer boundary is perfect electric conductor (PEC) which has tangential electric fields as zero. PEC will reflect electromagnetic waves for sure. Therefore, absorbing boundary condition will be needed in FDTD calculation.

In this dissertation, FDTD simulations will use the convolutional perfectly matched layer (CPML) to model the absorbing boundary condition [10]. As shown in Fig.1.3, CPML is used to absorb the outgoing EM waves in order to model free space calculation domain.

Figure 1.3 Illustration of CPML effect on EM wave reflection
1.2.3 Total Field and Scattered Field Model

To model the effect of scattering of objects from plane wave incidence, total-field/scattered-field (TF/SF) technique is applied in FDTD algorithm. As illustrated in Fig.1.4 (a), the calculation domain is divided into 3 regions. The CPML region absorbs the outgoing wave to model an infinite free space. The scattered field region only simulate the scattered electric/magnetic fields that caused by the plane wave incidence. Finally, the total field region, as shown inside the red box, presents the total field for the wave interaction with scattering objects.

Figure 1.4 (a) Total field/scattered field technique effects on EM wave scattering (b) Matlab simulation for a square PEC object
Fig. 1.4 (b) shows a Matlab simulation for a PEC object. In this two-dimensional simulation, the incident wave is a plane wave with Gaussian signature going through from lower left to upper right. The scattering object is denoted by the white square and has no field inside because its boundary is PEC. In the total field region, both incident wave and scattered field present. In the scattered field region, only the field scattered by the PEC square is observed. Beyond the scattered region is the CPML region where the EM fields are always decaying towards the outer PEC boundary and causes no reflection.

1.2.4 Periodic Structure

As stated in the previous section, electromagnetic metamaterials usually are periodic structures created by many identical resonant scattering objects. Periodic structure means infinite size in one or more directions. Considering the limited computer memory and the calculation efficiency, periodic boundary condition (PBC) will be needed to simulate metamaterials with periodic structures [11, 12, 13, 14].

1.2.5 FDTD Algorithm Acceleration

To speed up the simulation in modern FDTD technique, parallel computation has to be used so that multiple execute units can work at the same time. FDTD mainly use cubic cell meshing and calculates electrical and magnetic fields separately, which makes it very suitable for parallel computing. While the graphics processing unit (GPU) has become an integral part of today’s mainstream computing systems, we expect
dramatically increase in productivity by reducing our FDTD simulation time when it is running on GPU.

Another method to accelerate the simulation is to use subgrid so that only part of the computing region needs to be discretized with fine cell size. At the meanwhile, the subgrid method saves memories too.

1.3 Topics and Chapters to Cover

The following parts of the dissertation will discuss the problems mentioned above and will provide verification and simulation results for certain cases.

Chapter 2 shows the FDTD implementation of three common linear dispersive materials, namely Debye, Lorentz and Drude material. It also shows utilization in plasmonic array nano-antennas on layered substrates.

In Chapter 3, the topic of nonlinear material as well as gain material will be discussed.

In Chapter 4, the powerful general-purpose analog electronic circuit simulator SPICE is coupled with our FDTD engine, through the implementation of inter-process communication between these two interfaces of the software. Details will be discussed about the procedure of the implementation and verification. Then we investigate three cases of metamaterial application: metamaterial structure constructed from periodic resonant loops integrated with lumped capacitor elements, loop metamaterial with integrated non-Foster transistor configurations, and metasurface integrated with non-linear varactors.
The work of Chapter 5 and 6 is designated to accelerate the speed of the FDTD simulator. Chapter 5 studies parallel computing for FDTD technique. It introduces FDTD the cutting-edge area of GPU computing and aims to make FDTD engine fast and capable to simulate huge structures. The implementation of FDTD with OpenCL framework will be shown. In Chapter 5 we emphasize on the area of subgrid method in order to speed up the FDTD engine when simulating large structures with small-size elements.
Chapter 2

Dispersive Materials

When electromagnetic waves are interacting with the medium in which they are propagating, dispersion may be observed and it relates the wave properties such as frequency, velocity (phase and energy), and wavelength and so on. In a dispersive material, waves with different frequency would travel at different velocities. There are two classes of dispersive material: electric and magnetic. They are described by the permittivity and permeability, respectively. Dispersion could be linear or nonlinear. Linear dispersive material’s permittivity and/or permeability change with frequency. Nonlinear material’s permittivity and/or permeability change with the field intensity that applied to the material. Nonlinear dispersive material’s permittivity and/or permeability change with both frequency and the applied field intensity. Compared to these dispersive materials, gain material is active, which is able to magnify the amplitude of the propagating electromagnetic wave inside the material.

Piecewise Linear Recursive Convolution (PLRC) method [8, 15, 16] and Auxiliary Differential Equation (ADE) methods [17] are widely used methods for dispersive material simulation in FDTD. While they both have second order accuracy, the time-domain basis of ADE method makes modeling of arbitrary nonlinear dispersive media as well as gain material particularly attractive. The following sections discuss the
auxiliary differential equation (ADE) method and follow the equations from [8] for FDTD implementation.

2.1 ADE Method for Linear Dispersive Material

Here we address three linear dispersive materials: Debye, Lorentz and Drude materials. Since all three materials are dispersive, the permittivity is a function of frequency

\[ \varepsilon_r(\omega) = \varepsilon_\infty + \sum_{p=1}^{p} \chi_p(w) \]  

(2.1)

where \( \varepsilon_r(\omega) \) is the relative permittivity, \( \varepsilon_\infty \) is the relative permittivity at infinite frequency, \( p \) is the number of poles and \( \chi_p(w) \) is the susceptibility, which is the dispersive part of the relative permittivity.

Substituting Eq.(2.1) into Eq.(1.1), we got Ampere’s Law in time domain

\[ \nabla \times \mathbf{H} = \sigma \mathbf{E} + \frac{\partial (\varepsilon \mathbf{E})}{\partial t} = \sigma \mathbf{E} + \varepsilon_0 \varepsilon_\infty \frac{\partial \mathbf{E}}{\partial t} + \sum_{p=1}^{p} \mathbf{J}_p \]  

(2.2)

\[ \mathbf{J}_p = j\omega \varepsilon_0 \chi_p \mathbf{E} \]  

(2.3)

As can be seen for the above equations, for dispersive material we have one extra term \( \mathbf{J}_p \) compared to the simple dielectric case. Therefore Fig.1.2 does not apply anymore and the FDTD field update procedure needs to count in the new term \( \mathbf{J}_p \), as shown in Fig.2.1.
So, we need to develop a method to obtain the polarization current $J_p$. Different material will have different forms of $J_p$, we will discuss the details below for Debye, Drude and Lorentz materials.

### 2.1.2 Material with Multiple Debye Poles

Generally, dielectric relaxation is the momentary delay (or lag) in the dielectric constant of a material. Debye relaxation is characterized by a complex-valued, frequency-domain susceptibility function $\chi_p(\omega)$ that has one or more real poles at separate frequencies. The single pole model for Debye material is depicted as the following function:

$$\chi_p(\omega) = \frac{\varepsilon_{s,p} - \varepsilon_{\infty,p}}{1 + j\omega\tau_p} = \frac{\Delta\varepsilon_p}{1 + j\omega\tau_p}$$

(2.4)

where $\varepsilon_{s,p}$ is the static or zero-frequency relative permittivity, $\Delta\varepsilon_p = \varepsilon_{s,p} - \varepsilon_{\infty,p}$ is the change in relative permittivity due to the Debye pole, and $\tau_p$ is the pole relaxation time. For Debye media with multiple poles, the permittivity function is:
\[ \varepsilon_r(\omega) = \varepsilon_{\infty} + \sum_{p=1}^{P} \frac{\Delta \varepsilon_p}{1 + j\omega \tau_p} \]  \hfill (2.5)

\[ \bar{J}_p = \varepsilon_0 \Delta \varepsilon_p \left( \frac{j\omega}{1 + j\omega \tau_p} \right) \bar{E} \]  \hfill (2.6)

where \( \bar{J}_p \) is the phasor polarization current associated with each pole. By multiplying 1 + \( j\omega \tau_p \) on both sides, Eq.(2.6) becomes

\[ (1 + j\omega \tau_p) \bar{J}_p = \varepsilon_0 \Delta \varepsilon_p j\omega \bar{E} \]  \hfill (2.7)

Eq.(2.7) can be transformed by inverse Fourier transformation as

\[ J_p + \tau_p \frac{\partial J_p}{\partial t} = \varepsilon_0 \Delta \varepsilon_p \frac{\partial E}{\partial t} \]  \hfill (2.8)

This equation, together with Eq.(2.2), are the ADEs for \( J_p \) that makes the electric field update equation. To carry out the update, semi-implicit finite-difference scheme centered at \( n+1/2 \) is utilized for Eq.(2.8)

\[ J_p^{n+1} = k_p J_p^n + \beta_p \left( \frac{E^{n+1} - E^n}{\Delta t} \right) \]  \hfill (2.9)

where the coefficients are

\[ k_p = \frac{1 - \Delta t}{2\tau_p} ; \quad \beta_p = \frac{\varepsilon_0 \Delta \varepsilon_p \Delta t}{\tau_p} \]  \hfill (2.10)

Since the Ampere’s Law Eq.(2.2) is used to update the electric field and is centered at time \( n+1/2 \), the polarization current \( J \) needs to be represented at time \( n+1/2 \). This can be done averaging Eq.(2.9) at time \( n \) and \( n+1 \).
\[ J_{p}^{n+1/2} = \frac{1}{2} (J_{p}^{n} + J_{p}^{n+1}) = \frac{1}{2} \left[ (1+k_{p})J_{p}^{n} + \frac{\beta_{p}}{\Delta t}(E_{n+1}^{p} - E^{n}) \right] \] (2.11)

Then Ampere’s Law can be written as

\[ \nabla \times H_{n+1/2}^{p} = \varepsilon_{0} \varepsilon_{\infty} \frac{E_{n+1}^{p} - E^{n}}{\Delta t} + \sigma \frac{E_{n+1}^{p} + E^{n}}{2} \]

\[ + \frac{1}{2} \sum_{p=1}^{P} \left[ (1+k_{p})J_{p}^{n} + \frac{\beta_{p}}{\Delta t}(E_{n+1}^{p} - E^{n}) \right] \] (2.12)

From the above equation, electric field \( E \) can be updated from time step \( n \) to time step \( n+1 \) with the following equation

\[ E^{n+1} = \left( \frac{2\varepsilon_{0}\varepsilon_{\infty} + \sum_{p=1}^{P} \beta_{p} - \sigma \Delta t}{2\varepsilon_{0}\varepsilon_{\infty} + \sum_{p=1}^{P} \beta_{p} + \sigma \Delta t} \right) E^{n} \]

\[ + \left( \frac{2\Delta t}{2\varepsilon_{0}\varepsilon_{\infty} + \sum_{p=1}^{P} \beta_{p} + \sigma \Delta t} \right) \left[ \nabla \times H_{n+1/2}^{p} - \frac{1}{2} \sum_{p=1}^{P} (1+k_{p})J_{p}^{n} \right] \] (2.13)

To sum up the procedure, first Ampere’s Law Eq.(2.13) is used to update \( E \) field. Then Eq.(2.11) is used for polarization current \( J \) update. After that, the regular FDTD algorithm with Faraday’s Equation Eq.(1.2) is applied. The calculation procedure for electric dispersive material with multiple Debye poles is shown in Fig.2.2.

Figure 2.2 FDTD field update procedure for Debye material
Because we have extra terms for the calculation of $J$, we need more memory to store both the coefficients and the polarization current values for each cell. We still use the coefficients for the simple dielectric case. We just add the corresponding terms $\sum_{p=1}^{P} \beta_p$ and $\frac{1}{2} \sum_{p=1}^{P} (1+k_p)J_p$ for dispersive material, so that we don’t need to change the original FDTD code while adding new dispersive materials.

Up to now, the effect of Debye material with multiple poles has been fully considered into the electric field update equation. Compared to the normal dielectric, it’s easy to find out that, for this Debye material we only need to take care of the extra $J_p$ term. Then for material with different polarization current $J_p$, we can always use this method without worrying about the other parts of the FDTD algorithm. The implementations of Lorentz and Drude materials are carried out in a similar way. In the next chapter we will see that this also works even for nonlinear material and gain material.

### 2.1.3 Material with Multiple Lorentz Poles

The susceptibility function of Lorentz material is in the form of complex Lorentzian function, which has one or more pairs of complex-conjugate poles. A Lorentz material that has $p$ pole pairs can be described by the following relative permittivity function:

$$
\varepsilon_r(\omega) = \varepsilon_{\infty} + \sum_{p=1}^{P} \frac{\Delta \varepsilon_p \omega_p^2}{\omega_p^2 + 2j \omega \delta_p - \omega^2}
$$

(2.14)
where $\varepsilon_r(\omega)$ is the relative permittivity; $\varepsilon_\infty$ is the relative permittivity at infinite frequency; $p$ is the number of poles; $\Delta \varepsilon_p = \varepsilon_{r,p} - \varepsilon_{\infty,p}$, is the change in relative permittivity due to the Lorentz pole pair; $\omega_p$ is the frequency of the pole pair and $\delta_p$ is the damping coefficient.

To apply the Ampere’s Law of Eq.(2.2), we need to obtain the phasor polarization current, which is associated with the pole pair of the susceptibility function

$$J_p = j\omega \frac{\varepsilon_0 \Delta \varepsilon_p \omega_p^2}{\omega_p^2 + 2j\omega \delta_p - \omega^2} \vec{E} \quad (2.15)$$

To calculate the phasor polarization current in time domain, we first multiply $\omega_p^2 + 2j\omega \delta_p - \omega^2$ on each side of Eq.(2.15), then perform the inverse Fourier transformation on the frequency domain equation

$$(\omega_p^2 + 2j\omega \delta_p - \omega^2)J_p = j\omega \varepsilon_0 \Delta \varepsilon_p \omega_p^2 \vec{E} \quad (2.16)$$

$$\omega_p^2 J_p + 2\delta_p \frac{\partial J_p}{\partial t} + \frac{\partial^2 J_p}{\partial t^2} = \varepsilon_0 \Delta \varepsilon_p \omega_p^2 \vec{E} \quad (2.17)$$

Note that both electric field and the phasor polarization current are calculated in integer time steps, therefore by applying the finite-difference to the above partial differential terms, we have

$$\omega_p^2 J^n_p + 2\delta_p \frac{J^{n+1}_p - J^{n-1}_p}{2\Delta t} + \frac{J^{n+1}_p - 2J^n_p + J^{n-1}_p}{(\Delta t)^2} = \varepsilon_0 \Delta \varepsilon_p \omega_p^2 \frac{E^{n+1} - E^{n-1}}{2\Delta t} \quad (2.18)$$

Then the update equation for the phasor polarization current will be

$$J^{n+1}_p = \alpha_p J^n_p + \xi_p J^{n-1}_p + \gamma_p \frac{E^{n+1} - E^{n-1}}{2\Delta t} \quad (2.19)$$
Similar to the Debye material, the Ampere’s Law Eq. (2.2) is used to update the electric field and is centered at time \( n+1/2 \), the polarization current \( J \) needs to be represented at time \( n+1/2 \). This is done by time-averaging two terms at \( n \) and \( n+1 \)

\[
\begin{align*}
J_p^{n+1/2} &= \frac{1}{2} (J_p^{n+1} + J_p^n) \\
&= \frac{1}{2} \left[ (1+\alpha_p)J_p^n + \xi_p J_p^{n-1} + \frac{\gamma_p}{2\Delta t} (E_p^{n+1} - E_p^{n-1}) \right]
\end{align*}
\]  

Then the Ampere’s Law becomes

\[
\nabla \times H^{n+1/2} = \varepsilon_0 \varepsilon_r \frac{E^{n+1} - E^n}{\Delta t} + \sigma \frac{E^{n+1} + E^n}{2} + \frac{1}{2} \sum_{p=1}^P \left[ (1+\alpha_p)J_p^n + \xi_p J_p^{n-1} + \frac{\gamma_p}{2\Delta t} (E_p^{n+1} - E_p^{n-1}) \right]
\]  

After some mathematical manipulation, we have the following explicit update equation for electric field

\[
E^{n+1} = C_1 E^{n+1} + C_2 E^n + C_3 \left\{ \nabla \times H^{n+1/2} - \frac{1}{2} \sum_{p=1}^P \left[ (1+\alpha_p)J_p^n + \xi_p J_p^{n-1} \right] \right\}
\]  

where

\[
C_1 = \frac{1}{2} \sum_{p=1}^P \gamma_p \quad , \quad C_2 = \frac{2\varepsilon_0 \varepsilon_r - \sigma \Delta t}{D} \quad , \quad C_3 = \frac{2\Delta t}{D}
\]  

\[
D = 2\varepsilon_0 \varepsilon_r + \frac{1}{2} \sum_{p=1}^P \gamma_p + \sigma \Delta t
\]  

\[
\alpha_p = \frac{2 - \varepsilon_0^2 (\Delta t)^2}{\delta_p \Delta t + 1}, \xi_p = \frac{\delta_p \Delta t - 1}{\delta_p \Delta t + 1}, \gamma_p = \frac{\varepsilon_0 \varepsilon_r \omega_p^2 (\Delta t)^2}{\delta_p \Delta t + 1}
\]  

(2.20)
To sum up, Apmere’s Law Eq.(2.23) is used to update $E$ field. Then Eq.(2.19) is used for polarization current $J$ update. After that, the regular FDTD algorithm with Faraday’s Equation Eq.(1.2) is applied. Again, we need more memory to store both the coefficients and the polarization current value for each cell. Note that for Lorentz material we need to store more coefficients and polarization current values than Debye material, such as $J_p^{n-1}$ and $E^{n-1}$. Because each cell only has one type of material, we can reuse those memory defined for Debye material while adding extra terms, so as to save some memory. The calculation procedure for electric dispersive material with multiple Lorentz pole pairs is shown in Fig.2.3.

$$E^n \xrightarrow{\text{Eq.(1.2)}} H^{n+1/2} \xrightarrow{\text{Eq.(2.23)}} E^{n+1} \xrightarrow{\text{Eq.(1.2)}} H^{n+3/2} \xrightarrow{\text{Eq.(2.23)}} E^{n+2}$$

Figure 2.3 FDTD field update procedure for Lorentz material

2.1.4 Material with Multiple Drude Poles

Metals usually have high conductivity and are often seen as perfectly electric conductor at low frequencies. According to the electromagnetic theory, when the frequency of electromagnetic wave approaches to infinite, which means a wavelength of zero, everything is transparent to the wave, including metal. Thus, when frequency goes up to a certain level, metal does not have PEC property anymore and we need new model to describe the physics. Drude model is such a model to account for this change for metal
above certain frequency, such as optical frequency. It describes the electron motion of metal and the related electromagnetic wave interactions. Multi-pole Drude material is characterized by the following permittivity function:

\[
\varepsilon_r(\omega) = \varepsilon_{\infty,p} - \sum_{p=1}^{P} \frac{\omega_p^2}{\omega^2 - j\omega\gamma_p}
\]  

(2.26)

where \(p\) is the number of pole; \(\varepsilon_{\infty,p}\) is the relative permittivity at infinite frequency; \(\omega_p\) is the frequency of the \(p\)th pole and \(\gamma_p\) is the inverse of the \(p\)th pole relaxation time. The frequency domain polarization current for \(p\)th pole is

\[
\mathbf{J}_p = -j\omega\varepsilon_0 \frac{\omega_p^2}{\omega^2 - j\omega\gamma_p} \mathbf{E}
\]

(2.27)

To calculate the phasor polarization current in time domain, we first multiply \(\omega^2 - j\omega\gamma\) on each side of Eq.(2.27), then perform the inverse Fourier transformation on the frequency domain equation

\[
(\omega^2 - j\omega\gamma_p) \mathbf{J}_p = -j\omega\varepsilon_0 \omega_p^2 \mathbf{E}
\]

(2.28)

\[
\frac{\partial^2 \mathbf{J}_p}{\partial t^2} + \gamma_p \frac{\partial \mathbf{J}_p}{\partial t} = \varepsilon_0 \omega_p^2 \frac{\partial \mathbf{E}}{\partial t}
\]

(2.29)

After integration on both side we have

\[
\frac{\partial \mathbf{J}_p}{\partial t} + \gamma_p \mathbf{J}_p = \varepsilon_0 \omega_p^2 \mathbf{E}
\]

(2.30)

Note that both electric field and the phasor polarization current are calculated at integer time steps, therefore by applying the finite-difference to the above partial differential terms, we have
\[
\frac{J_{p}^{n+1} - J_{p}^{n}}{\Delta t} + \gamma_{p} J_{p}^{n+1} + J_{p}^{n} = \varepsilon_{0} \alpha_{p}^{2} \frac{E_{p}^{n+1} + E^{n}}{2}
\] (2.31)

Then the update equation for the phasor polarization current will be

\[
J_{p}^{n+1} = k_{p} J_{p}^{n} + \beta_{p} (E^{n+1} + E^{n})
\] (2.32)

where

\[
k_{p} = \frac{1 - \gamma_{p} \Delta t / 2}{1 + \gamma_{p} \Delta t / 2}, \quad \beta_{p} = \frac{\varepsilon_{0} \omega_{p}^{2} \Delta t / 2}{1 + \gamma_{p} \Delta t / 2}
\] (2.33)

Similar to the Debye and Lorentz material, the Ampere’s Law Eq.(2.2) is used to update the electric field and is centered at time \( n+1/2 \), the polarization current \( J \) needs to be represented at time \( n+1/2 \). This is done by time-averaging two terms at \( n \) and \( n+1 \)

\[
J_{p}^{n+\frac{1}{2}} = \frac{1}{2} (J_{p}^{n+1} + J_{p}^{n}) = \frac{1}{2} \left[ (1 + k_{p}) J_{p}^{n} + \beta_{p} (E^{n+1} + E^{n}) \right]
\] (2.34)

Then the Ampere’s Law becomes

\[
\nabla \times H^{n+\frac{1}{2}} = \varepsilon_{0} \varepsilon_{\infty} \frac{E^{n+1} - E^{n}}{\Delta t} + \sigma \frac{E^{n+1} + E^{n}}{2}
\]

\[
+ \frac{1}{2} \sum_{p=1}^{P} \left[ (1 + k_{p}) J_{p}^{n} + \beta_{p} (E^{n+1} + E^{n}) \right]
\] (2.35)

Then we have the following explicit update equation for electric field

\[
E^{n+1} = \begin{cases} 
\frac{2 \varepsilon_{0} \varepsilon_{\infty} - \Delta t \sum_{p=1}^{P} \beta_{p} - \sigma \Delta t}{2 \varepsilon_{0} \varepsilon_{\infty} + \Delta t \sum_{p=1}^{P} \beta_{p} + \sigma \Delta t} & E^{n} \\
\frac{2 \Delta t}{2 \varepsilon_{0} \varepsilon_{\infty} + \Delta t \sum_{p=1}^{P} \beta_{p} + \sigma \Delta t} & \nabla \times H^{n+\frac{1}{2}} = \frac{1}{2} \sum_{p=1}^{P} (1 + k_{p}) J_{p}^{n}
\end{cases}
\] (2.36)
To sum up, Apmere’s Law Eq.(2.36) is used to update E field. Then Eq.(2.32) is used for polarization current $J$ update. After that, the regular FDTD algorithm with Faraday’s Equation Eq.(1.2) is applied. Compared to Lorentz material, we have less parameters to store. Then we don’t need any new items to define Drude material. The calculation procedure for electric dispersive material with multiple Drude pole is shown in Fig.2.4.

Figure 2.4 FDTD field update procedure for Drude material

2.2 Implementation and Verifications

The previous sections discussed FDTD calculation procedure for electric dispersive material. Due to the beautiful symmetry of Maxwell’s equations, these methods can also be applied to magnetic dispersive materials, resulting in similar update equations for magnetic field.

Some electromagnetic applications require simulation with infinite scale structures, such as half space dielectric. In these cases, in order to model infinite structure, we need to extent the structure into the PML region to avoid reflection. As a result, the general procedure for FDTD calculation would be, $E \rightarrow PML \rightarrow H \rightarrow PML \ldots H \rightarrow PML$. When we introduce the polarization current $J$ for dispersive materials, we need to
calculate $J$ based on $E$ field. Here special notice need to be taken that, we don’t obtain $J$ immediately after the $E$ update (as shown in Fig.2.5 by the red crossing signs). Instead, we obtain $J$ from $E$ after the PML region calculation. This way we ensure that the $E$ fields used to update $J$ are always the latest ones. The field update procedure is shown in Fig.2.5.

![Figure 2.5 FDTD field update procedure for dispersive materials with PML calculation](image)

We make two simulations to verify our implementation with ADE method for dispersive materials. In the first case, we calculate the transmission coefficient for an infinite slab with thickness of 60mm that is characterized by Drude model, as shown in Fig.2.6. Due to the property of the Drude model, it is an ENG material in the frequency of interest.

![Figure 2.6 Drude material slab with electric negative property](image)
As can be seen from Fig.2.6, the slab is periodic in y&z directions and has a thickness of 60mm in x direction. The Drude material is characterized by \( \varepsilon_{\infty,p} = 1; \omega_p = 2\pi \times 4 \times 10^9 \text{ (rad/s)}; \gamma_p = 0.001 \omega_p \). A plane wave with Gaussian signature incidents in –x direction with electric field and magnetic field in z and y directions, respectively.

Since we upgraded our FDTD code from PLRC method, we’ll compare our ADE results with the ones from PLRC calculation [16]. Fig.2.7 and Fig.2.8 show the magnitude and phase for the transmission coefficient of the slab. Both of them match well with the results from PLRC method.

![Graph showing the magnitude of transmission coefficient for the ENG medium](image)

Figure 2.7 Magnitude of transmission coefficient for the ENG medium
The second example is for Lorentz material. In this case, a plane wave incidents to a half-space medium characterized by Lorentz material with three pole pairs. The half-space effect is modeled by extending Lorentz material to the PML region. The Lorentz medium has parameters $\varepsilon_{\infty, p} = 1; \varepsilon_s = 3$ and three Lorentzian resonances in the optical range: $(f_1 = 2 \times 10^{14} \text{Hz}, \delta_1 = 0.5 f_1), (f_2 = 4 \times 10^{14} \text{Hz}, \delta_2 = 0.5 f_2), (f_3 = 6 \times 10^{14} \text{Hz}, \delta_3 = 0.5 f_3)$. We’ve obtained the analytical results for the reflection coefficient magnitude and the comparison with ADE method is shown in Fig.2.9. Again, the figure shows a perfect match between the two curves. And moreover, ADE method apparently presents result that is closer to the analytical one than the PLRC method.
2.3 Application and Numerical Results

At microwave frequencies, in order to achieve a high gain antenna performance, one needs to use the concept of array antennas where different radiators are tailored in unique arrangements to control the field’s amplitudes and phases in the far-field spectrum enabling directive emission. The following application shows FDTD simulation with ADE method applied to the modeling of plasmonic array nano-antennas on layered substrates and the results are taken from paper [18].

The configuration of a layered substrate plasmonic particle-antenna is shown in Fig.2.10. The geometry consists of sub-wavelength nano core-shells implanted above an
infinitely long (in x and y directions) semiconductor substrate where layers of plasmonic or dielectric materials are coating the semiconductor surfaces. Here, the plasmonic particles are spherical core-shells with dielectric cores $\varepsilon_{rc}$ and plasmonic coatings $\varepsilon_{rs}$. The inner and outer radii for the nano-shell are $r_1$ and $r_2$, respectively; the thickness for the $n^{th}$ layer substrate is $d_n$. The use of plasmonic material as the coating allows greater control on scattering characteristics of the nano core-shells [19].

![Diagram of layered substrates with plasmonic array nano-antennas on top](image)

Figure 2.10 Layered substrates with plasmonic array nano-antennas on top [18]

In our simulation, the nanoparticles will operate at resonance frequency. It is well know that to tune the resonant frequency for a simple metallic nano-particle, one should change the total size of the particle, while in many applications it is required to maintain a desired size. However, for nano-shell particles we have one more degree of freedom. Thus the scattering resonance of a plasmonic concentric core-shell structure can be obtained at desired frequencies by tuning the $(r_1/r_2)$ radii ratio [7].
After obtaining the resonance frequency for the nano-shells, we investigate both the radiation pattern and the near field for the Yagi-Uda type optical nanoantenna structure as depicted in Fig.2.11. In the structure, the core of the shell is dielectric SiO$_2$ and the shell is defined by Drude model

$$\varepsilon_{rs} = 1 - \frac{1.513 \times 10^{32}}{\omega(\omega + i7.9 \times 10^{13})} \quad (2.37)$$

The working frequency is 650THz (\(\lambda_0 = 461 \text{nm}\)), the radii ratios for reflector and director are \(r_{1r}/r_{2r} = 0.75\) and \(r_{1d}/r_{2d} = 0.65\), respectively. In this scenario, since the optical dipole is polarized in the z direction, the induced dipoles on the nano core-shells are also polarized in the z direction.

![Figure 2.11 Nano-shell particles with dielectric as core and Silver as shell [18]](image)

The radiation patterns for the 2-element Yagi-Uda nano-shells are obtained using our FDTD engine and are demonstrated in Fig.2.12 (a) and (b) in xy and yz planes, respectively. The blue curve in the figure comes from a theoretical analysis based on Green’s function by viewing the particles as induced electric dipoles which are related to the total electric field upon that particle by a polarizability factor.
Despite that FDTD characterizes the actual plasmonic core-shell structure whereas in theoretical model approximate dipole modes are used, a good comparison is observed. In the results, the radiation patterns plotted are the radiated power normalized to the maximum value of the radiated power of the optical dipole source in the absence of the nano core-shells and the layered substrate.

![Normalized radiated power](image)

**Figure 2.12** Normalized radiated power [18] obtained (a) in xy plane, and (b) in yz plane for nano-shells depicted in Fig.2.11

Beside the radiation pattern, the near-fields based on theory and FDTD techniques are also compared, as shown in Figs.2.13(a) and (b). Good comparisons are observed. It is worth highlighting that, FDTD characterizes the complete core-shell particles while in the theoretical model we approximate them with dipole elements located at the core-shells’ centers. Therefore, the near field characteristics of the antenna obtained by applying the theoretical approach show the field behavior of dipoles. This implies that,
the electric field has local maxima at the location of the dipoles (centers of the core-shells), while the FDTD provides the dipole behavior outside the core-shells. Therefore, one does not observe the local maxima in the center of the core-shells. Nevertheless, considering the small size of particles and the fact that we are interested in the field performance outside the particles, this is a very good approximation as observed from the comparisons.

Figure 2.13 Magnitude of Ez(dB) for the nanoantenna in Fig.2.11 in xy (z=0) plane and yz (x=0) plane. (a) Theory, (b) FDTD [18]
Chapter 3

Nonlinear and Gain Materials

3.1 Nonlinear Dispersive Material

Nonlinear material’s permittivity and/or permeability change with respect to the field intensity that applied to the material. Nonlinear dispersive material’s permittivity and/or permeability change with both frequency and the applied field intensity. This section follows the ADE method presented by Fujii et al. [20] as discussed in [8].

3.2 Algorithm for Nonlinear Dispersive Material

Again, we start from the fundamental equation Ampere’s Law

\[ \frac{\partial D}{\partial t} = \nabla \times H \] (3.1)

Nonlinear material usually comes with linear dispersion. Now, considering both linear and nonlinear part of the permittivity function, the constitutive relation is expressed as

\[ D = \varepsilon_0 \varepsilon_\infty E + P^L + P^{NL} \] (3.2)

where \( P^L \) and \( P^{NL} \) are the linear and nonlinear part of the dielectric polarization, respectively. The linear dispersion has been discussed in the previous section and we can
use those methods as before. For the material we discuss here, we assume the linear polarization $P^L$ includes two parts: Debye dispersion and Lorentz dispersion. For the nonlinear polarization $P^{NL}$, there are two third-order terms included: Kerr and Raman dispersive nonlinearity.

### 3.2.1 Contribution of Linear Debye Polarization

As discussed in the previous sections, a single pole model for Debye material is depicted as the following function:

$$\chi_p(\omega) = \frac{\varepsilon_{s,p} - \varepsilon_{\infty,p}}{1 + j\omega\tau_p} = \frac{\Delta\varepsilon_p}{1 + j\omega\tau_p}$$

(3.3)

where $\varepsilon_{s,p}$ is the static or zero-frequency relative permittivity, $\Delta\varepsilon_p = \varepsilon_{s,p} - \varepsilon_{\infty,p}$ is the change in relative permittivity due to the Debye pole, and $\tau_p$ is the pole relaxation time.

For convenience sake, we don’t calculate the polarization current $J$ here. Instead we will deal with the electric polarization $P$, even they are essentially the same.

$$\vec{P}_{Debye} = \frac{\varepsilon_0\Delta\varepsilon_p}{1 + j\omega\tau_p} \vec{E}$$

(3.4)

To calculate the phasor polarization in time domain, we first multiply $1 + j\omega\tau_p$ on each side of Eq.(3.4), then perform the inverse Fourier transformation on the frequency domain equation

$$P_{Debye} + \tau_p \frac{\partial P_{Debye}}{\partial t} = \varepsilon_0\Delta\varepsilon_p E$$

(3.5)
To carry out the update, semi-implicit finite-difference scheme centered at n+1/2 is utilized for Eq.(3.5)

\[ P_{\text{Debye}}^{n+1} = a_{\text{Debye}} P_{\text{Debye}}^n + b_{\text{Debye}} (E^{n+1} + E^n) \]  

(3.6)

where

\[ a_{\text{Debye}} = \frac{2\tau_p - \Delta t}{2\tau_p + \Delta t} \quad b_{\text{Debye}} = \frac{\varepsilon_0 \Delta \varepsilon_p \Delta t}{2\tau_p + \Delta t} \]

### 3.2.2 Contribution of Linear Lorentz Polarization

As discussed in the previous Chapter section, the pole pair model for Lorentz material is depicted as the following function:

\[ \chi_p(\omega) = \frac{\Delta \varepsilon_p \omega_p^2}{\omega_p^2 + 2j\omega \delta_p - \omega^2} \]  

(3.7)

where \( \varepsilon_r(\omega) \) is the relative permittivity; \( \varepsilon_\infty \) is the relative permittivity at infinite frequency; \( p \) is the number of poles; \( \Delta \varepsilon_p = \varepsilon_{s,p} - \varepsilon_{\infty,p} \), is the change in relative permittivity due to the Lorentz pole pair; \( \omega_p \) is the frequency of the pole pair and \( \delta_p \) is the damping coefficient.

Again, we don’t calculate the polarization current \( J \) here. Instead we will deal with the electric polarization \( P \)

\[ \vec{P}_{\text{Lorentz}} = \frac{\varepsilon_0 \Delta \varepsilon_p \omega_p^2}{\omega_p^2 + 2j\omega \delta_p - \omega^2} \vec{E} \]  

(3.8)
To calculate the phasor polarization in time domain, we first multiply \( \omega_p^2 + 2j\omega\delta_p - \omega^2 \) on each side of Eq.(3.8), then perform the inverse Fourier transformation on the frequency domain equation

\[
\omega_p^2 P_{\text{Lorentz}} + 2\delta_p \frac{\partial P_{\text{Lorentz}}}{\partial t} + \frac{\partial^2 P_{\text{Lorentz}}}{\partial t^2} = \epsilon_0 \Delta \epsilon_p \omega_p^2 E
\]

To carry out the update, semi-implicit finite-difference scheme centered at \( n+1/2 \) is utilized for Eq.(3.9)

\[
P_{\text{Lorentz}}^{n+1} = \frac{2 - \omega_p^2 (\Delta t)^2}{\delta_p \Delta t + 1} P_{\text{Lorentz}}^n + \frac{\delta_p \Delta t - 1}{\delta_p \Delta t + 1} P_{\text{Lorentz}}^{n-1} + \frac{\epsilon_0 \Delta \epsilon_p \omega_p^2 (\Delta t)^2}{\delta_p \Delta t + 1} E^n
\]

### 3.2.3 Contribution of Third-Order Nonlinear Polarization

In time domain, the third-order nonlinear polarization is denoted by the convolution between the third-order susceptibility function \( \chi^{(3)}(t_1,t_2,t_3) \) and the electric field \( E \). The Born-Oppenheimer approximation provides a relatively simple model to account for the electron response [21]

\[
P_{NL}(t) = \epsilon_0 \chi^{(3)}(t) E(t) \int_{-\infty}^{\infty} g(t-t') E^2(t') dt'
\]

where \( \chi^{(3)} \) is the strength of the third-order nonlinearity, \( g(t) \) is the causal response function and is normalized as

\[
\int_{-\infty}^{\infty} g(t) dt = 1
\]

We are considering the third-order nonlinearities of Kerr term with non-resonant virtual electronic transitions on the order of 1fs or less and transient Raman term of
scattering effect. Since the causal response function \( g(t) \) is a combination of the two term, then we have [22]

\[
g(t) = \alpha \delta(t) + (1 - \alpha) g_{\text{Raman}}(t)
\]

(3.13)

where \( \alpha \) is a constant real number in the range \([0,1]\) to account for the relative strengths of the Kerr and Raman terms; \( \delta(t) \) is a Dirac delta function corresponding to Kerr nonlinearity; the Raman scattering term \( g_{\text{Raman}}(t) \) is given by

\[
g_{\text{Raman}}(t) = \left( \frac{\tau_1^2 + \tau_2^2}{\tau_1^2 \tau_2^2} \right) e^{-t/\tau_2} \sin(t/\tau_1) U(t)
\]

(3.14)

Now let’s consider the Kerr and Raman terms separately. For the Kerr nonlinearity, the polarization contribution is given by [21]

\[
P_{\text{Kerr}}(t) = \varepsilon_0 \chi_0^{(3)} (E(t))^3 \int_{-\infty}^{\infty} \alpha \delta(t-t') E^2(t') dt'
\]

(3.15)

Due to the property of Dirac delta function, the integral in the above equation is very straight forward. The update equation is

\[
P_{\text{Kerr}}(t) = \alpha \varepsilon_0 \chi_0^{(3)} (E^{n+1})^3
\]

(3.16)

Now for the Raman nonlinearity polarization, its contribution is denoted by the time domain convolution

\[
P_{\text{Raman}}(t) = \varepsilon_0 E(t) \left[ \chi_{\text{Raman}}^{(3)}(t)(t) \ast E^2(t) \right]
\]

(3.17)

where

\[
\chi_{\text{Raman}}^{(3)}(t) = (1 - \alpha) \chi_0^{(3)} g_{\text{Raman}}(t)
\]

(3.18)

Then, to calculation the convolution, we introduce an auxiliary variable

\[
S(t) = \chi_{\text{Raman}}^{(3)}(t) \ast E^2(t)
\]

(3.19)
and the time domain polarization for Raman nonlinearity can be written

\[ P_{\text{Raman}}(t) = \varepsilon_0 E(t) S(t) \] (3.20)

Since time domain convolution means multiplication in frequency domain, the Fourier transform of Eq.(3.19) is

\[ S(\omega) = \chi_{\text{Raman}}^{(3)}(\omega) \cdot \mathbb{F}\left[ E^2(\omega) \right] \] (3.21)

where the response function is

\[ \chi_{\text{Raman}}^{(3)}(\omega) = \frac{(1 - \alpha) \chi_0^{(3)} \omega_{\text{Raman}}^2}{\omega_{\text{Raman}}^2 + 2 j \omega \delta_{\text{Raman}} - \omega^2} \] (3.22)

and the parameters are

\[ \omega_{\text{Raman}} = \sqrt{\frac{\tau_1^2 + \tau_2^2}{\tau_1^2 \tau_2^2}}; \delta_{\text{Raman}} = \frac{1}{\tau_2} \] (3.23)

Substituting Eq.(3.22) into Eq.(3.21), we have

\[ S(\omega) = \frac{(1 - \alpha) \chi_0^{(3)} \omega_{\text{Raman}}^2}{\omega_{\text{Raman}}^2 + 2 j \omega \delta_{\text{Raman}} - \omega^2} \cdot \mathbb{F}\left[ E^2(\omega) \right] \] (3.24)

To calculate the auxiliary term \( S \) in time domain, we first multiply \( \omega_{\text{Raman}}^2 + 2 j \omega \delta_{\text{Raman}} - \omega^2 \) on each side of Eq.(3.24), then perform the inverse Fourier transformation on the frequency domain equation

\[ \omega_{\text{Raman}}^2 S + 2 \delta_{\text{Raman}} \frac{\partial S}{\partial t} + \frac{\partial^2 S}{\partial t^2} = (1 - \alpha) \chi_0^{(3)} \omega_{\text{Raman}}^2 E^2 \] (3.25)

Note that both electric field \( E \) and the auxiliary variable \( S \) are calculated in integer time steps, therefore by applying the finite-difference to the above partial differential terms, we have
This is the update equation for the auxiliary variable $S$. With the knowledge of $S$ and using Eq.(3.20), the update equation for Raman polarization is

$$P_{\text{Raman}}^{n+1}(t) = \varepsilon_0 E^{n+1} S^{n+1}$$  \hspace{1cm} (3.27)

Now, back to the constitutive relation Eq.(3.2), we have obtained all the terms to update the electric flux density $D$

$$D^{n+1} = \varepsilon_0 \varepsilon_\infty E^{n+1} + P_{\text{Debye}}^{n+1} + P_{\text{Lorentz}}^{n+1} + P_{\text{Kerr}}^{n+1} + P_{\text{Raman}}^{n+1}$$  \hspace{1cm} (3.28)

It seems that we are ready to calculate electric field $E$ from the above equation. However, we note that both the Kerr and Raman terms are not linear function of electric field $E$. The Raman polarization $P_{\text{Raman}}^{n+1}$ has a three-order term $E^{n+1}(E^n)^2$, while the Kerr polarization $P_{\text{Kerr}}^{n+1}$ on the right hand side has a three-order term of $(E^{n+1})^3$. As a result, this is a transcendental equation for electric field, and the following Newton Iteration method [23] is used to obtain $E^{n+1}$

$$E^{<m+1>} = \frac{D^{n+1} - a_{\text{Debye}} P_{\text{Debye}}^{n+1} - b_{\text{Debye}} E^n - P_{\text{Lorentz}}^{n+1}}{\varepsilon_0 \varepsilon_\infty + b_{\text{Debye}} + \alpha \varepsilon_0 \chi_0^{(3)} (E^{<m>})^2 + \varepsilon_0 S^{n+1}}$$  \hspace{1cm} (3.29)

for $m=0,1,2,...$. $E^{<m>}$ is the $m^{th}$ approximation for $E^{<m+1>}$ in Newton iteration method. The initial value of $E^{<m+1>}$ can just be $E^n$. Based on numerical experiments, the number of iteration needed is determined by the property of the material, as well as the time step corresponding to the operation frequency. Based on our the numerical experiments, 5 or
even 3 iterations would be accurate enough for the electric field update Eq.(3.28) for most cases.

3.2.4 Implementation and Numerical Results for Nonlinear Dispersive Material

The discussed algorithm for nonlinear dispersive material calculation is outlined in Fig.3.1. As shown below, it starts from a time step with known electric field $E$ and ends back to $E$ to make one full FDTD cycle. First, PML is applied to the whole simulation domain. Then, with the PML-updated electric field $E$, magnetic field $H$ is updated with the regular FDTD calculation. Moreover, the auxiliary variable $S$, as well as the two linear dispersive polarization terms $P_{\text{Debye}}$ and $P_{\text{Lorentz}}$, are also obtained from the latest $E$ field. Then Ampere’s Law Eq.(3.1) is used to update the electric flux density $D$. Finally, Newton iteration method is utilized to update the electric field.

![Figure 3.1 FDTD field update procedure for nonlinear material](image)

Figure 3.1 FDTD field update procedure for nonlinear material
To show the validity of the above implementation, a material with single linear Lorentz polarization and third-order Kerr and Raman nonlinearity is investigated. The material has the following linear Lorentz dispersive parameters: \( \varepsilon_s = 5.25 \); \( \varepsilon_\infty = 2.25 \); \( \omega_p = 4 \times 10^{14} \text{ sec}^{-1} \); \( \delta_p = 2 \times 10^6 \text{ sec}^{-1} \). And the three-order nonlinear parameters are:

\[
\chi_0^{(3)} = 7 \times 10^{-2} (V/m)^2; \alpha = 0.7; \tau_1 = 12.2 \text{ fs}; \tau_2 = 32 \text{ fs}.
\]

In this simulation, a pulsed optical signal source with Gaussian signature is propagating into this half-space material. The half-space effect is modeled by extending the material into the PML region. Comparison is made between linear dispersive material and nonlinear material (\( \chi_0^{(3)} \) is set to 0) as shown in Fig.3.1 and Fig.3.2, respectively. In these two figures, the x axis is time and y axis is magnitude of the electric field. Red curve of wave form is snapshot at \( t=487 \text{ fs} \) while the green one is taken at \( t=973 \text{ fs} \). As can be seen from Fig.3.2, a small precursor pulse is formed in the wave front of the soliton due to the dispersion and nonlinearity of the material. Apparently, small precursor pulse has a faster group velocity than the main Gaussian pack. Due to the three-order nonlinearity, the precursor pulse has a frequency that is about three times of the main Gaussian packed wave.
Figure 3.2 Gaussian wave pack after propagating 55um and 126um in linear Lorentz material

Figure 3.3 Gaussian optical wave pack after propagating 55um and 126um in nonlinear dispersive material
3.3 Optical Gain Material

In this section we will show the implementation of optical gain material according to the ADE-FDTD method described in [24]. This gain material is not laser material that usually addressed at the atomic level. Instead, it is defined at the macroscopic scale and has saturable, frequency-dependent gain at optical frequencies.

3.4 Algorithm for Optical Gain Material

To address the algorithm for optical gain material, we will describe the problem in one-dimension with field components $E_z$ and $H_y$ propagating along $x$ direction through this electric gain medium. The one dimensional Faraday equation becomes

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu_0} \frac{\partial E_z}{\partial x}$$

(3.30)

and Ampere’s Law becomes

$$J_z + \varepsilon_0 \varepsilon \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x}$$

(3.31)

The conductivity function that will be discussed here is given by

$$\sigma(\omega) = \frac{J_z(\omega)}{E_z(\omega)} = s \left[ \frac{\sigma_0 / 2}{1 + j(\omega - \omega_0)T_2} + \frac{\sigma_0 / 2}{1 + j(\omega + \omega_0)T_2} \right]$$

(3.32)

where $I_s$ is the saturation intensity, $s \equiv (1 + I / I_s)^{-1}$ is the saturation coefficient, $\sigma_0$ is related to the peak value of the gain. According to [24], population inversion happens when $\sigma_0$ is negative, providing gain.
### 3.4.1 Algorithm

To solve the one-dimensional Maxwell’s equations, we first make the inverse Fourier transformation of Eq.(3.32)

\[
(1 + \omega_0^2 T_2^2) J_z + 2 T_2 \frac{\partial J_z}{\partial t} + T_2^2 \frac{\partial^2 J_z}{\partial t^2} = s \sigma_0 E_z + s \sigma_0 T_2 \frac{\partial E_z}{\partial t}
\]

By defining an auxiliary variable \( F_z \)

\[
F_z \equiv \frac{\partial J_z}{\partial t}
\]

Eq.(3.33) can be written as

\[
(1 + \omega_0^2 T_2^2) J_z + 2 T_2 F_z + T_2^2 \frac{\partial F_z}{\partial t} = s \sigma_0 E_z + s \sigma_0 T_2 \frac{\partial E_z}{\partial t}
\]

Using time domain difference at time step \( n+1/2 \), the update equations can be obtained by solving the coupled differential equations (3.31), (3.34) and (3.35)

\[
F_z |_{n+1} = A_1 \left( H_y |_{n+1/2} - H_y |_{n-1/2} \right) + A_2 E_z |_n + A_3 J_z |_n + A_4 F_z |_n
\]

\[
J_z |_{n+1} = J_z |_n + \frac{\Delta t}{2} \left( F_z |_{n+1} + F_z |_n \right)
\]

\[
E_z |_{n+1} = E_z |_n - \frac{\Delta t}{2 \epsilon} \left( J_z |_{n+1} + J_z |_n \right) + \frac{\Delta t}{\epsilon \Delta x} \left( H_y |_{n+1/2} - H_y |_{n-1/2} \right)
\]

where

\[
A_1 = \frac{4 \Delta t s(i) \sigma_0 (\Delta t + 2 T_2)}{B \Delta x}; A_2 = \frac{4 \epsilon s(i) \sigma_0 \Delta t}{B}
\]

\[
A_3 = -\frac{4 \Delta t \left[ 2 \epsilon (1 + \omega_0^2 T_2^2) + s(i) \sigma_0 (\Delta t + 2 T_2) \right]}{B}
\]
\[
A_1 = -\frac{8\varepsilon T_2 (\Delta t - T_2)}{B} - \frac{(\Delta t)^2}{B} \left[ 2\varepsilon (1 + \omega_0^2 T_2^2) + s(i)\sigma_0 (\Delta t + 2T_2) \right]
\]

(3.41)

\[
B = 8\varepsilon T_2 (\Delta t - T_2) + (\Delta t)^2 \left[ 2\varepsilon \varepsilon_0 (1 + \omega_0^2 T_2^2) + s(i)\sigma_0 (\Delta t + 2T_2) \right]
\]

(3.42)

\[
s(i) = \left[ 1 - I(i)/I_s \right]^{-1}; I(i) = 0.5cn \left( E_c \right)_\text{peak}^2
\]

(3.43)

The function \( I(i) \) is always determined by the up to date maximum electric field in running history for the current cell. Since \( I_s \) is usually a big constant number, \( s(i) \) will be zero for linear material.

### 3.4.2 Implementation and Numerical Results for Optical Gain Material

The discussed algorithm for macroscopic optical gain dispersive material calculation is outlined in Fig.3.4. As shown below, it starts from a time step with known electric field \( E \) and ends back to \( E \) to make one full FDTD cycle. First, PML is applied to the whole simulation domain. Then, with the PML-updated electric field \( E \), magnetic field \( H \) is updated with the regular FDTD calculation. After that, before we use Eq.(3.36) to update the auxiliary variable \( F \), we need to check if we have a new peak for the current cell. If we do have a new peak, then the coefficients \( A1, A2, A3, A4 \) and \( B \) need to be updated based on the latest \( s(i) \) value. Then we can update \( F \) using Eq.(3.36), followed by \( J \) update using Eq.(3.37). Finally, electric field \( E \) will be obtained from Eq.(3.38)
To show the validity of the above implementation, a linear gain medium of gallium arsenide (GaAs) is studied. The material characterized by the following parameters: \( n = 3.59 \), \( T_2 = 0.07 \text{ ps} \), \( \sigma_0 = -5000 \text{ S/m} \), \( \varepsilon = 12.8881 \), \( \omega_0 = 2.1165 \times 10^{15} \), \( \lambda_0 = 0.89 \mu m \), \( I_x = 6.52 \times 10^4 \text{ W/m}^2 \).

In this simulation, a Gaussian pulse centered at \( \omega_0 \) is used as the source. We record the time waveform of electric field \( E \) at two locations \( x_1 \) and \( x_2 \) that are separated by a distance \( l = x_2 - x_1 = \lambda_0 / n \). Then we can obtain the propagation factor \( \gamma = \alpha + j \beta \) by taking the ratio of the DFTs of the two waveforms. Fig.3.5 and Fig.3.6 compare the amplification factor and the phase factor for the propagation parameter \( \gamma \), respectively. As can be seen, the FDTD results denoted by the red line fit well with the exact solutions in both cases.
Figure 3.5 Comparison for amplification factor of optical gain material

Figure 3.6 Comparison for phase factor of optical gain material
Chapter 4

FDTD Coupling with SPICE

In recent years, the subject of metamaterials integrated with electronic circuits, nonlinear elements, and non-Foster transistor configurations have been drawing significant attention in research communities. Metamaterials are typically periodic structures created by array of resonant elements which are small and stationary compared to the wavelength of electromagnetic wave. The combination of metamaterials with circuit elements can bring novel and tunable characteristics to these materials. For instance combining loop-based metamaterials with varactors provides a means to electronically tune the performance by applying a DC voltage or a high power signal without modifying the structure [25, 26, 27, 28]. Alternatively, a non-Foster transistor configuration can be introduced to enhance the metamaterial bandwidth [29]. Active components can be also integrated with transmission line metamaterials to compensate the loss [30], and finally metasurface can be integrated with circuit elements to tailor their reflection phase performance [31].
4.1 Lumped Circuit Model for FDTD

Along with the progress of computational electromagnetics, FDTD technique has been able to incorporate circuit elements in its time domain simulation. As a full-wave simulation engine, FDTD has ability to calculate the electromagnetic fields accounting for circuit effects, as well as field effects. This is an advantage compared to traditional circuit simulator working with chip level. This section implements the method in [32] for lumped circuit elements.

First, the physics of FDTD being able to simulate circuit elements lies in the basic Ampere’s Law

\[ \nabla \times H = \varepsilon \frac{\partial E}{\partial t} + J_c \]  

(4.1)

where \( J_c = \sigma E \) is the electric conduction current.

Suppose we have a lumped circuit element that has a contribution of current \( J_L \), then Ampere’s Law becomes

\[ \nabla \times H = \varepsilon \frac{\partial E}{\partial t} + J_c + J_L \]  

(4.2)

Similar to the electric conduction current \( J_c \), as long as the lumped circuit current \( J_L \) can be denoted by electric field \( E \), Eq.(4.2) will be an explicit or implicit function of \( E \) and can be solved to update the electric filed.

In circuit model, we usually make calculation corresponding to the voltage \( V \) and current \( I \). In three dimensional cases, assume we have one lumped circuit element along \( z \)
direction in a FDTD cell as shown in Fig.4.1, the total cell current \( I \) and the local current density \( J_L \) are related by

\[
J_L = \frac{I_L}{\Delta x \Delta y}
\]  

(4.3)

Similar connection between electric field \( E \) and voltage \( V \) along the cell edge is characterized by

\[
E_L = \frac{V}{\Delta z}
\]  

(4.4)

Figure 4.1 Unit FDTD cell illustration for lumped circuit current calculation

Since for a certain lumped circuit element, the voltage-current relation is known, then the relation between current density \( J_L \) and electric field \( E \) can be found, such as
\( J_L = f(E) \). Substituting this relation into Ampere’s Law leads to the update equation for electric field.

To sum up, we have the relation between current density \( J \) and total current \( I \) based on the FDTD Yee cell grid, as well as the relation between the electric field \( E \) and voltage \( V \) along the cell edge. Then, for a certain circuit element we extract the relation between the voltage \( V \) and the current \( I \) across it. This circuit \( V/I \) relation connects the current density \( J \) and electric field \( E \). Finally, \( J \) is denoted by \( E \) and is substituted into Ampere’s Law and to obtain the update equation for FDTD calculation.

Fig. 4.2 shows the procedure of FDTD calculation incorporating circuit elements. Note that in the figure, because the differential equation Eq. (4.4) is centered at half time steps, it relates the voltage \( V \) with both current time step \( E \) and next time step \( E \). After the update for \( E \), magnetic field calculation will simply follow the regular FDTD procedure. The figure also implies that we can always model a circuit element device as long as we can extract its voltage current relation.

Figure 4.2 FDTD field update procedure for circuit elements
4.1.1 Resistive Voltage Source

To simulate a resistive voltage source, suppose the internal resistance inside the voltage source is $R_s$ and the voltage of the source is $V_s$. Then according to the resistive voltage source definition, we have

$$ I_L = \frac{V_{cell}}{R_s} + \frac{V_s}{R_s} \quad (4.5) $$

This is the relation between current and voltage. Then consider the three dimensional case that the resistive voltage source is along $z$ direction in a FDTD cell, and note that Ampere’s Law is centered at half time step, Eq.(4.5) becomes

$$ I_{z}^{n+1/2} = \frac{\Delta z}{2R_s} \left( E_{z}^{n+1} + E_{z}^{n} \right) + \frac{V_s^{n+1/2}}{2R_s} \quad (4.6) $$

$$ J_L = \frac{I_z^{n+1/2}}{\Delta x \Delta y} \quad (4.7) $$

In Eq.(4.6) the relation between electric field $E$ and voltage $V$ is used. Eq.(4.7) relates current density $J$ to the total current $I$. Substitute all these relations into Ampere’s Law, finally the electric filed update equation will be

$$ E_{z}^{n+1} = \left( \frac{1 + \frac{\Delta t \Delta z}{2R_s \varepsilon_0 \Delta x \Delta y}}{1 + \frac{\Delta t \Delta z}{2R_s \varepsilon_0 \Delta x \Delta y}} \right) E_{z}^{n} + \left( \frac{\Delta t}{\varepsilon_0} \frac{\Delta t \Delta z}{2R_s \varepsilon_0 \Delta x \Delta y} \right) \left( \nabla \times H \right)_{z}^{n+1/2} $$

$$ + \left( \frac{\Delta t}{R_s \varepsilon_0 \Delta x \Delta y} \right) \frac{\Delta t \Delta z}{2R_s \varepsilon_0 \Delta x \Delta y} V_s^{n+1/2} \quad (4.8) $$
4.1.2 Diode

Semiconductor diodes are commonly used in circuits systems. The p-n junction connected to two electrical terminals makes the diode to allow an electric current to pass only in one direction. A simple model to describe the current and voltage relations is given by

\[ I_d = I_s (e^{qV_d/kT} - 1) \]  \hspace{1cm} (4.9)

where \( I_s \) is the reverse bias saturation current; \( q \) is the electron charge; \( V_d \) is the voltage across the diode; \( k \) is the Boltzmann’s constant; \( T \) is the absolute temperature.

For the implementation we use a numerically stable 3D FDTD algorithm with the semi-implicit update strategy for \( E \) field [32]

\[ E_{z}^{n+1/2}_{i,j,k} = \frac{1}{2} \left( E_{z}^{n}_{i,j,k} + E_{z}^{n+1}_{i,j,k} \right) \]  \hspace{1cm} (4.10)

The diode voltage \( V_d \) is in the exponent, so is the electric field \( E \). Substituting Eq.(4.10) into Ampere’s Law results in a transcendental equation [32]

\[ E_{z}^{n+1}_{i,j,k} = E_{z}^{n}_{i,j,k} + \frac{\Delta t}{\varepsilon_0} \left( \nabla \times H \right)_{z}^{n+1/2}_{i,j,k} \]

\[ - \frac{I_s \Delta t}{\varepsilon_0 \Delta x \Delta y} \left\{ \exp \left[ -q \left( E_{z}^{n}_{i,j,k} + E_{z}^{n+1}_{i,j,k} \right) \frac{\Delta z}{2kT} \right] - 1 \right\} \]  \hspace{1cm} (4.11)

Similar to the transcendental equation in nonlinear material calculation, Eq.(4.11) is solved using Newton’s method. Based on the numerical experiments, 4 iterations would be accurate enough for the solution. Because of fast changing current nature of diode, small cell size such as 0.001 wavelength is desired to get a smoothed output.
Fig. 4.3 and Fig. 4.4 show a test circuit and its results from the above FDTD algorithm for diode simulation, respectively. It can be seen for the output of diode voltage in Fig. 4.4 that the method is stable with a voltage up to 10v across the diode.

Figure 4.3 FDTD circuit simulation with resistive voltage source and diode

Figure 4.4 Voltage output for diode element
One big problem with FDTD simulation of circuit elements using the method from previous section is that, it is capable to model only a few simple cases. For example, the above diode current equation only has several parameters that come from an approximated model. However, commercial software usually has tens of parameters included. Direct FDTD simulations with more complex semiconductor devices, such as transistors and CMOS, involves very complicated model and are not recommended.

To use the power of other circuit simulators, as well as fully take the advantage of various libraries, we will discuss the coupling between FDTD and SPICE.

4.2 Introduction to FDTD-SPICE Model for Metamaterial Simulation

The complexity of the structure requires a powerful computational technique to bridge the physics of electromagnetic metamaterial with circuit configuration. In the following sections of this chapter, the Finite Difference Time Domain (FDTD) technique is combined with SPICE-like software to provide a capable numerical technique for time domain full wave analysis of the composite structure. The technique is applied to the design and simulation of metamaterials integrated with circuit elements. Broadband performance is obtained with the focus on bandwidth and tunable behavior engineering.

Section 4.3 provides the details of FDTD-SPICE hybridization and shows the validation through some examples. In section 4.4, the following three applications are explored, a) loop-based metamaterial integrated with lumped capacitors to tune the resonance performance, b) metamaterial integrated with non-Foster transistor
configuration for bandwidth manipulation, and c) a metasurface combined with varactors featuring tunable reflection phase.

4.3 FDTD-SPICE Computational Model

4.3.1 Hybridization of FDTD with SPICE

As one of the major computational electromagnetic modeling schemes, Finite Difference Time Domain technique can tackle problems by providing a full wave solution in time. It is a capable method for modeling a variety of scientific and engineering problems [8, 33]. On the other hand, the general-purpose analog electronic circuit simulator, SPICE [34], can solve complex circuit topologies in time domain. It makes great sense to combine FDTD with SPICE to feature a powerful numerical engine for solving metamaterials integrated with circuit configurations.

FDTD uses Maxwell’s equations (1) and (2) in time domain to recursively calculate the electromagnetic fields at each position in three-dimensional space.

\[
\frac{\partial E}{\partial t} + J(E) = \nabla \times H \quad (4.12)
\]

\[
\frac{\partial H}{\partial t} + M(H) = \nabla \times E \quad (4.13)
\]

While both equations can provide the coupling between FDTD and SPICE according to reference [8], the first equation is used in this section.

Consider a FDTD unit cell in three dimensional spaces, with the edge length of unit cell denoted by \( dl (dl=dx=dy=dz) \), in uniform grids, the area of one face of the unit
cell by $A$ ($A=dl\times dl$ for uniform grids). Multiplying $A$ on both sides of Eq.(4.12) results in the following equation

$$
\varepsilon^* \frac{A}{dl} \frac{d(E^*dl)}{dt} + A^* J(E) = A^* \nabla \times H
$$

(4.14)

Then we note that in the above equation, $\varepsilon$ has a unit as $F/m$, $dl$ has a unit of $m$, thus $\varepsilon^* \frac{A}{dl}$ has the unit of Farad, which is the unit of capacitor. For the next terms, $E^* dl$ has the unit of voltage; $A^* J(E)$ has the unit of Ampere, the same as current; $A^* \nabla \times H$ also has the unit of Ampere. And physically, $E^* dl$ is voltage and $A^* J(E)$ is current, and so on.

So, with the above simple manipulations, Ampere’s Law of Eq.(4.12) can be written as [35]

$$
C \frac{dV}{dt} + I(V) = I
$$

(4.15)

where the whole current on the right hand side is

$$
I = A^* \nabla \times H
$$

(4.16)

And the voltage across the element is

$$
V = E^* dl
$$

(4.17)

Now look at Eq.(4.15), the first term on the left hand side is indeed the definition for the current of a capacitor; the second term on the left hand side is the current across the simulated device; on the right hand side we have a current source, which is the total current of the two. As a result, this equation actually describes a circuit model as shown
in Fig.4.5. In this circuit, \( C = \varepsilon * \frac{A}{dl} \) is a capacitance that abstracted from the model (note that here \( C \) is not the intrinsic lattice capacitance of FDTD cell). \( I(V) = A*J(E) \) is the current through the SPICE element. This model is the foundation for the coupling between FDTD and SPICE.

![Circuit model for Ampere’s Law equation that connects FDTD and SPICE](image)

With this model, we divide the FDTD simulation domain into two parts: the regular part, and the cells that contain SPICE elements. For the regular part, FDTD simply use the Faraday equation Eq.(1.2) to update magnetic field. For the cells containing SPICE elements, instead of using Eq.(1.1) to update the electric field, SPICE can carry out Eq.(4.15) and obtain the voltage across the element and then pass it to FDTD program. From the voltage, it is easy to obtain \( E \) field afterward. Fig.4.6 describes the procedure of this calculation.
To determine the voltage across the corresponding circuit element, SPICE software needs to run a transient simulation for a period of $\Delta t$, which should be the same as FDTD time step. Then in the next time step, regular FDTD method will update all the fields except the cell that has SPICE element. For the SPICE element, SPICE program needs to run another $\Delta t$, starting from where it stopped last time. Before each run of SPICE, FDTD sends data that contains circuit information to SPICE. After each time step of SPICE simulation, it passes results back to FDTD. In this way, the FDTD and SPICE engines run alternatively until the end of the whole simulation time. With the help of SPICE, FDTD is ready to simulate any complex integrated configuration as long as it is provided in SPICE’s library. Since conventional FDTD can only model very simple circuit elements, SPICE function adds a big advantage to FDTD.

The SPICE software used here is SPICE3f5. It has an interactive mode that can receive commands from and send results back to the interface. It also has “stop” and “resume” commands that can set a breakpoint after running some time steps in a transient simulation and then resume it. These features make it very suitable to be connected to FDTD engine.
To exchange data between FDTD and SPICE, inter-process communication should be developed. Fig. 4.7 presents a complete picture of our FDTD-SPICE model. When a simulation begins, FDTD main program calls “fork” function which spawns two processes: the son process that will execute SPICE using “exec” function, and the parent process which is the original FDTD process itself. Two sockets: Socket[0] and Socket[1] are created in the main program before it forks so that both of the two children have access to the sockets. Here these sockets act like two-way pipe, where FDTD process
sends SPICE commands to Socket[0] and reads results from Socket[1], while SPICE process reads message from Socket[0] and sends results back to Socket[1]. FDTD process will be waiting for SPICE outputs while SPICE process is running and vice versa. These two processes then communicate through the two sockets at each time step until the end of the simulation.

While DC solution is needed in SPICE for many cases but not very easy to be obtained with FDTD, literature [34, 36] addressed DC solution problem by separating the AC and DC components, which is helpful for the metasurface simulations of diode varactors with bias that presented in the following sections.

Our FDTD code has been written in a very powerful manner and it has many features such as modeling periodic structures and linear dispersive metamaterials [16]. By applying auxiliary differential equation (ADE) method [20], it can also model nonlinear dispersive materials as well as gain materials. The absorbing wall is based on CPML [10] where one can truncate the computational domain very close to the structure. The code has been applied successfully to a variety of metamaterials composites [33] [37, 38, 39].

4.3.2 Test Cases for FDTD-SPICE Model

Two cases are studied to validate the FDTD-SPICE model. All simulations in this section use uniform gridding and are run on Sun Ultra45 workstation with 2GHz CPU and 4GB memory.

In the first case as depicted in Fig. 4.8, one parallel RLC circuit is connected to a Gaussian pulse voltage source in order to obtain the impedance. There are about 8000
time steps in the calculation with each time step $\Delta t=0.2\,ps$, cell size $\Delta x=0.15\,mm$.

Including CPML boundary, the whole computational domain is modeled by $26 \times 26 \times 26$ Yee’s cells. The FDTD process takes 456 seconds, while, since the circuit is simple, SPICE process takes only 10 seconds. Fig. 4.9 shows both the FDTD-SPICE results and the analytical calculation for the real and imaginary parts of the impedance $Z$, which agree well with each other.

Figure 4.8 RLC circuit configuration

Figure 4.9 RLC circuit impedance $Z$ behavior.
In the second case as shown in Fig. 4.10, one BFG403W transistor from Discrete Semiconductor is utilized in a single frequency simulation with a sinusoidal voltage source. The BJT has a saturation current $I_S=5.554\,\text{mA}$ and the forward active current gain is $\beta_F=145$. The calculation has 18000 time steps with each time step $\Delta t=54.82\,\text{fs}$ and cell size $\Delta x=30\,\mu\text{m}$. Because of the complex property of BJT circuit, the cell size and time step are much smaller than those used in previous simulation. The whole computational domain has a size of $25\times37\times32$ Yee’s cells. FDTD and SPICE processes take 1734 seconds and 41 seconds, respectively. Fig. 4.11 shows the comparison for the waveform of voltage across resistor $R_2$ in time domain, between FDTD-SPICE and complete SPICE simulation. An excellent agreement is demonstrated.

Figure 4.10 Transistor circuit configuration
Figure 4.11 Output voltage waveform in time. The FDTD-SPICE simulation matched very well with pure SPICE simulation.

4.4 FDTD-SPICE Model Applications on Metamaterials

This section presents applications of FDTD-SPICE method to some recent metamaterial designs, namely loop-based composite metamaterial with capacitors, metamaterial integrated with non-Foster circuits and nonlinear elements loaded metasurface. In all of the cases, uniform gridding, perfectly matched layers, and periodic boundary conditions are used. Since they all deal with plane wave incidence, total field/scattered field method is also utilized [37].
4.4.1 Metamaterial Loops Integrated with Passive Elements

A periodic structure made from metallic loops can artificially create a medium with permeability characteristics (positive and negative) [39]. The metamaterial structure to be simulated here is shown in Fig. 4.12. It is a four layer structure in x direction where each layer has loops periodic in y-z directions and the loops are terminated to capacitor $C_p$. The unit loop dimension is 4.8mm and 1.8mm in x and z directions, respectively. The excitation is a plane wave that has $E_z/H_y$ components and propagates in $-x$ direction with a Gaussian signature. A transmission line analogy can be obtained for plane wave propagation through the medium [19], which is embedded with a loop that has a self-inductance $L_p$ and is terminated to a lumped capacitor $C_p$. By changing the circuit elements ($L_p$ and/or $C_p$) in the equivalent transmission line model, the effective parameter of the medium can be tuned, as well as the property of the metamaterial.

Figure 4.12 Metamaterial constructed from array of loops terminated to capacitors $C_p$
FDTD-SPICE method is applied to model the structure and obtain the transmission coefficient as shown in Fig. 4.13. The loops are modeled in FDTD and the capacitors in SPICE. There are 15700 time steps in the calculation with each time step $\Delta t=0.2\text{ps}$, cell size $\Delta x=0.15\text{mm}$. The whole computational domain has a size of $216\times8\times20$ Yee’s cells. FDTD and SPICE processes take 1669 seconds and 1319 seconds, respectively. Because of the multiple-port inter-process communication, SPICE process needs considerable simulation time, which may be improved by introducing batch data communication between processes.

![Figure 4.13 Transmission coefficient of metamaterial constructed from array of loops terminated to capacitors $C_p$](image)

Figure 4.13 Transmission coefficient of metamaterial constructed from array of loops terminated to capacitors $C_p$
Fig. 4.13 shows 4 simulation results. The red dots represent results from pure FDTD simulation with \( C_p = 2.1 \) pF (\( C_p \) is also modeled in FDTD). Then the blue line comes from FDTD-SPICE simulation with the same parameter, and it matches well with red dots, which show the accuracy of our model. Changing the capacitor \( C_p \) allows tuning the resonance and as well as the equivalent permeability of the metamaterial. Fig. 4.13 also shows the transmission coefficients for the metamaterial loops terminated to \( C_p = 4.2 \) pF and \( C_p = 8.4 \) pF lumped capacitors, depicted in black and magenta respectively. Doubling capacitance would half the resonance frequency. As can be seen from Fig. 4.13, the resonance frequency is shifted from 1.1 GHz down to 0.78 GHz and 0.55 GHz respectively, according the value of capacitors.

4.4.2 Integration with Non-Foster Circuit Elements

Non-Foster elements have recently attracted significant interests due to their novel properties in offering negative impedance performance and successfully engineering the frequency dispersion characteristics of antennas, transmission lines, and metamaterials towards the goals of interests [40] [41] [42] [43]. A non-Foster element can be realized using a Negative Impedance Converter (NIC) circuit by translating a given positive load into a negative one. NIC circuit is made from transistor elements and one would need a comprehensive modeling scheme to characterize them while they are integrated with metamaterials. It is also worth mentioning that the design of NIC circuit involves several challenging issues such as noise, biasing and stability and special care must be taken to obtain a wideband performance for non-Foster circuit. The NIC stability (which is not the
subject of this chapter) has also been checked with the help of pole-zero analysis to ensure that the system has absolutely no pole in the Right-Half-Plane (RHP) of the complex S-plane.

Here we explore the integration of NIC circuit with metamaterial loops [29] and its FDTD-SPICE simulation to investigate the bandwidth enhancement of the structure. A two-layer loop-based composite metamaterial with capacitors as shown in Fig.4.12 is utilized. The loops are terminated to capacitors $C_p$ and inductors $L(4\text{nH})$. By integrating NIC designs offering $-2\text{nH}$ inductance, we reduce the effect of the positive inductance and enhance the bandwidth. The NIC circuit used in this section is based on Linvill’s model [44], and it is an example of Voltage Inversion NIC (VINIC), as shown in Fig.4.14.

![Figure 4.14 Basic non-Foster element: NIC configuration based on Linvill's design, and equivalent circuit model for the loop-based metamaterial integrated with NIC negative inductor](image-url)

Figure 4.14 Basic non-Foster element: NIC configuration based on Linvill's design, and equivalent circuit model for the loop-based metamaterial integrated with NIC negative inductor
The circuit consists of two N-type BJTs and a lumped passive inductor $L_1=2nH$ as the load. After adding the non-Foster circuit to the loop, the equivalent circuit model [39] for the transmission line is depicted in Fig.4.14, where the NIC configuration is included in a negative inductor $L_n$.

The simulation has 11600 time steps with each time step $\Delta t=0.274$ps, cell size $\Delta x=0.15$mm. The whole computational domain has a size of $128\times 8\times 20$ Yee’s cells. FDTD and SPICE processes take 686 seconds and 133 seconds, respectively. Because of the complexity of the NIC circuit model, SPICE needs more time than the test examples from previous section. The results are shown in Fig. 4.15.

Figure 4.15 Transmission coefficient for metamaterial of loops terminated with $i$: $C_p=0.13pF$, $L=4nH$; $ii$: $C_p=0.2pF$, $L=4nH$, NIC circuit $L_{\text{total}}=2nH$; $iii$: $C_p=0.2pF$, lumped inductor $L=2nH$
Here three cases are performed. Firstly, metamaterial with loop terminated to capacitor \( C_p = 0.13\text{pF} \) and lumped inductor \( L = 4\text{nH} \) is simulated. As depicted by the blue curve in Fig. 4.15, it has a resonance around 2.4GHz. Secondly, to broaden the bandwidth a NIC circuit \( L_n = -2\text{nH} \) is added to inductor \( L = 4\text{nH} \), making the whole inductance \( 2\text{nH} \). In order to make a fair comparison, the capacitor is made \( C_p = 0.2\text{pF} \) so that the two simulations have the same resonance. As can be observed from the red curve in Fig. 4.15, the metamaterial integrated with non-Foster element has a wider bandwidth than the case without it (blue curve). Increasing the value of \( -L_n \) can enable a wider bandwidth performance (as has been comprehensively detailed in [5]). Lastly, to validate the results of the second case, NIC circuit \( L_n = -2\text{nH} \) together with inductor \( L = 4\text{nH} \) are replaced by one lumped inductor \( L = 2\text{nH} \). As shown in Fig. 4.15, the black curve follows the behavior of the second case.

At the end we want to mention that, the reason we have considered a positive \( 4\text{nH} \) inductor in series with the \( -2\text{nH} \) non-Foster element is to make sure the time-domain SPICE modeling is stable (the equivalent total inductance is positive, \( +2\text{nH} \)).

### 4.4.3 Metasurface Integrated with Varactor Diodes

Metasurfaces have great potential for making low profile antennas [45] and achieving an electronically tunable reflection phase is of remarkable interest in this regard. This section presents the simulation for a structure of metasurface integrated with varactors. The metasurface used in this section is similar to the one in [31], which is designed to act as an artificial magnetic conductor (AMC) and uses active NIC circuits as
negative elements to increase the bandwidth. Here varactor diodes are integrated in the
metamaterial and makes the metasurface tunable in resonance frequency.

Fig. 4.16 shows a metasurface structure consisting of periodic square metal
patches and PEC ground plane with dielectric slab in between. It also presents the unit
cell of the metasurface loaded with varactor diodes. The structure is periodic in y and z
directions and the incidence plane wave is –x direction. The FDTD-SPICE method is
applied to simulate the structure. There are 7000 time steps in the calculation with each
time step $\Delta t=0.953$ ps, cell size $\Delta x=0.5$ mm. The whole computational domain has a size
of $45 \times 21 \times 21$ Yee’s cells. FDTD and SPICE processes take 386 seconds and 8 seconds,
respectively.

Figure 4.16 Metasurface structure with periodic square metal patches and its unit cell that loaded
with varactor diodes
Fig. 4.17 shows the reflection coefficient phase for the metasurface without varactor diodes, as well as those with diodes that have different biases. The green line, as a reference frequency response, represents the one without varactor diode, while the others are for results that have diode with different bias.

![Reflection phase for the metasurface](image)

Figure 4.17 Reflection phase for the metasurface. Controlling the bias of the diode tunes the resonance frequency

The varactor diode model used here is Skywork’s SMV1232-079 Hyperabrupt Tuning Varactors [46]. When the diode is added to the structure, it increases the total capacitance therefore decrease the resonance frequency of the metasurface. As can been seen from Fig. 4.18, typical capacitance values of the varactor diode decrease when the
reverse bias voltage increases. The capacitance is a maximum while there is no reverse bias voltage. Thus for reverse bias with smaller voltage it should give larger shift in resonance frequency, which agrees well with the simulation results.

Figure 4.18 Performance data for SMV1232 Diode Varactor

4.5 FDTD-SPICE Modeling Summary

This chapter develops a hybridized FDTD-SPICE model and applies it to the simulations of some novel metamaterials integrated with electronic circuits. The Maxwell’s equations are linked with SPICE through the interchange between electromagnetic fields and voltages across the SPICE element nodes. Inter-process communication technique with socket is used as it is crucial for FDTD and SPICE to
exchange data at proper running time. Diagrammatic flowchart is discussed that demonstrates the detail of the execution process as well as the data exchanging with the help of sockets. To show the validity of the model, two simple cases that have analytical solution or pure SPICE results are investigated. Then, three kinds of metamaterial are studied and modeled by the FDTD-SPICE engine. First, loop-based metamaterial with lumped capacitor is investigated where changing the capacitance can vary the resonance frequency. Then, integration of metamaterial with non-Foster circuit element, which provides negative impedance and increases the bandwidth, is studied and characterized. Lastly, metasurface simulation with integrated non-linear varactor elements shows a tunable reflection phase property for this metamaterial. The technique is very powerful and its accuracy is validated well. We expect our scheme to be of great engine for comprehensive simulation of metamaterials integrated with active and nonlinear circuits, and allow exploring new artificial materials combined with a lot of potential complex circuits.

4.6 Acknowledgment

This work was supported in part by the U.S. Office of Naval Research (ONR) Grant No. N00014-10-1-0264. The author would like to thank Allen Taflove from Northwestern University for his kind discussion on the FDTD-SPICE model theory, and acknowledge Vincent Thomas from Los Alamos National Laboratory for his helpful information on SPICE and sockets programming.
Chapter 5

GPU Computing for FDTD

5.1 Parallel Computing for FDTD

FDTD modeling ranges from lower radio frequency antenna analysis to visible light applications such as nano-size plasma and photonic crystals. As the frequency goes up, the wavelength goes down, thus same structure requires more cells in the discretization. This means that we need large array to store the variables, as well as powerful calculating elements to tackle these huge data and heavy computation.

According to Moore’s law [47], the number of transistors on integrated circuits doubles approximately every 18 months. To meet the high-performance computing needs, we can always find the new generation of computer architecture with higher clock frequency central processing unit (CPU) and larger memories. In past few years, Moore’s law was still in effect in prediction of the number of transistors on integrated circuits. However, due to the increasing processor power consumption, the increase of clock frequency for processors found its physical constrains. With the heat generation be the bottleneck for increasing clock frequency of processors, parallel computing will be the option for high performance FDTD engine.
5.2 Overview of GPU Computation

Graphics processing units (GPU) are processors that heavily optimized for computer graphics processing. Modern general-purpose computing on graphics processing units (GPGPU) has extended the function of GPUs to perform computation that usually done by CPU. In this section, we don’t want to dig deep on the GPU platform environments, device architectures, memory model and parallel execution. Instead, we will show the basic concept of GPU parallel computation and then present our scheme for FDTD implementation.

Parallelism is the most important reason that GPU can provide faster calculation than CPU. Consider a simple vector addition for three arrays that each has 16 integer elements, a serial C language implementation would be

\[
\text{for(int } i=0; i<16; ++i) \{
C[i] = A[i] + B[i];
\}
\]

With single thread on a single core CPU, the above code repeats 16 times. Suppose each addition takes one time cycle, then the serial code takes 16 time cycles, as show in Fig. 5.1 by the red thread procedure. With 4 threads running on 4 CPU cores, which is common in modern computers, each only need to run 4 additions and require 4 time cycles in total, as denoted by the orange thread procedure in Fig. 5.1. With GPU, the calculation could be further accelerated. 16 threads can be created for the 16 additions at the same time, resulting in a total running time of one time cycle. As long as there’s no data dependency between the array elements, parallel calculation can be applied. Real
computation analysis needs much more considerations, such as processor clock rate, processor cache architecture, memory size, memory clock rate, PCI bus bandwidth and so on. This vector addition still clearly shows the power of GPU parallel computation. Moreover, modern GPUs have thousands of cores available for massively multi-threaded computation. For example, the AMD HD 7970 GPU [48] has 2048 stream processors on one device while NVidia GeForce GTX 680 [49] has 1536 CUDA cores. Both of them provide very promising architecture for parallel computing.

Figure 5.1 Acceleration by parallel computation
To accelerate the simulation process, GPU is probably one of the best choices for FDTD. Firstly, FDTD algorithm is computational expensive. The main overhead for the simulation is floating point arithmetic operations, not those logical operations. This makes GPU computing very suitable for FDTD simulation. Secondly, there are two main calculations in FDTD: electric field update and magnetic field update. Each field is only related to the fields in the cells that next to it. It is ideal for parallel computing since it doesn’t have much data dependency. Last but not least, to deal with very large structure, the FDTD computing region can be divided into several sub regions. Each sub region only needs to exchange boundary information with its neighbor, which is suitable for GPU computing with multiple devices that further parallelizes the already parallel processes.

There are two major programming languages for GPU coding: CUDA and OpenCL. CUDA is built by NVidia specifically for their GPUs while OpenCL is a framework that can execute across heterogeneous platforms consisting of CPUs, GPUs and other processors. In this work we use OpenCL as our programming language.

5.3 GPU Implementation for FDTD

The OpenCL implementation for FDTD is one of the collaborative projects between Professor David Kaeli’s group and Professor Hossein Mosallaei’s group at Northeastern University. The OpenCL programming in this section was mainly done by Zhongliang Chen from Professor David Kaeli’s group. The author acknowledges Zhongliang Chen and Professor David Kaeli for their assistance on this section. The
original version of OpenCL code utilizes a single GPU device to accelerate the FDTD engine. Here we will address the implementation scheme for this work.

In OpenCL, the functions that will run on GPU devices are called kernels [50]. Each device has a command queue. Kernels, as well as commands for data transfer such as writing to and reading from a buffer, are enqueued into a command queue to request execution on GPU. Even GPU runs fast, not all the executions are suitable for GPU. For FDTD, we only put those computing intensive functions on GPU, including EM fields update, PML calculation and so on.

Fig. 5.2 shows the FDTD calculation procedure for OpenCL implementation. To make it simple, we only include the basic calculation steps for electric and magnetic field update. At the beginning of the simulation, we initialize the $E$, $H$, $J$, $M$ fields, as well as the coefficients for the update. Then we setup the environment for GPU device and transfer the initial data to GPU. Once data is ready on GPU, we will start the parallel calculating process. After one FDTD time cycle on GPU, if time $t$ hasn’t reached the end, then we will pass data from GPU to CPU and do some data processing on CPU. This data processing is necessary, since some operations must be done on CPU, such as writing intermediate results to a file on hard disk. Other operations like logic intensive process may also be run on CPU in order to get optimized overall performance. Next step, CPU will pass those processed data back to GPU and continue the next cycle until the time ends. When the calculation is done, some file manipulation and I/O related post-processing, such printing results, will be carried out on CPU.
The above single device version OpenCL code has been successfully developed and testified with sample simulations. With a single GPU device for a structure of 100 cells in x, y and z directions, it delivers 15x speedup on NVIDIA GeForce GTX 480 over Intel Core i7-920 @ 2.67GHz with SSE optimizations, as shown in Fig. 5.3.
5.4 Design Consideration and Optimization

Generally speaking, optimization is much harder than implementation. On one hand, developers need to know very well about the FDTD algorithm, including the data dependency, computation burden, memory allocation and IO procedure for the simulation structure, to optimize OpenCL code for GPU device. On the other hand, understanding the potential advantages of different hardware features, the OpenCL virtual models for different hardware architectures and how the virtual models image to real devices are key knowledge for the optimization. In this section, we share some thoughts on OpenCL implementation design and optimizations for FDTD algorithm.

Figure 5.3 Speedup of OpenCL implementation for FDTD simulation compared to CPU version
The first optimization concern would be the communication between CPU and GPU. Note that all traffic between the CPU and GPU must come through the PCI Express bus, which is significantly slower than access to DRAM and on-chip buffers. Thus, the data transfer in Fig. 5.2 between CPU and GPU are very time consuming in both ways. Data transfer may be the bottleneck for some applications that requires frequent data communication between the host and GPU devices. In FDTD algorithm, it stores fields at different time steps for both single frequency and multi-frequency simulations. Considering the output files are usually not big, one option is that we don’t write these outputs to files on hard disk. Instead, we could store them on global memory and write them to hard disk after the calculation is done.

In the OpenCL memory model, there are 4 memory types. From the slowest to the fastest they are global, constant, local and private memories. To fully use the advantage of fast memories, we will store as much variables as we can on the local and private memories. However, the sizes of these memories are small compared to global memory. We usually can’t put all the variables on them. Thus only the most frequently used variables shall be stored in the fast memories.

Knowing that the fast memories are very limited, we can reduce the number of variables that stored in global memory so as to decrease the date fetch time. For any FDTD simulation, we only have several possible coefficients, depending on the number of materials we have. Those coefficients are not relevant to the location of the cells and they don’t change during the whole simulation except for the optical gain material. Since the gain material is not frequently used compared to dispersive and nonlinear material,
we may optimize the coefficients variable on the memory behalf. For example, instead of allocating big three-dimensional arrays of coefficients for each cell in the whole computational region, we actually only need to allocate several constant coefficient values based on the materials we have in the structure of simulation. This coefficients deduction applies to both E/H fields and J/M fields.
Chapter 6

FDTD Subgrids

6.1 Subgrids Models in FDTD

The first step to carry out for a numerical method in engineering simulation is usually the discretization. In FDTD, the numerical technique is based on space discretization with three-dimensional unit cells called Yee cells. The whole algorithm, including the calculation and numerical stability, is related to the cell size which is usually defined according to the simulation frequency or wavelength.

In real simulations, the number of cells in a wavelength ranges from 20 to 200, or even more for special simulations like nonlinear dispersive material and circuit elements. In the area of metamaterial simulation, the structures are often of sub-wavelength size and provide a homogeneous electromagnetic property by interacting with the wave. That means, for the metamaterial structure simulation itself, wavelength as well as cell size is related to the structure size. For example, small cell size is used for small structures while big cell size used for large structures. Hence, the ratio of cell size and the wavelength does not change much between different simulations.

However, the above situation will change when we apply metamaterials to other objects for certain applications. For example, suppose we have a metamaterial antenna working at 30GHz. The wavelength is 1cm and the FDTD cell size may be 0.1mm. Since
metamaterial structure is sub-wavelength in size, we don’t have any issue simulating the metamaterial structure itself. But, if we want to apply it to a cellphone, we may have problem. The size of cellphone does not change with respective to the operating frequency of its antenna; even the antenna itself changes size for different frequencies. The size of the cellphone is only determined by the size of human hands. So, when we want to simulate the whole cellphone with metamaterial antenna included, the simulation domain may be too big to use a small cell size. And we can’t use bigger cell size either, because the cell size is determined by the smallest element of the structure, which in this case is the dimension of metamaterial. This is the situation when we have problem with uniform grids: we have both very large and very small dimensions for structures in model, compared to the simulation wavelength.

In another word, the problem arises when the key dimensions in the simulation range over several orders of magnitude. That is, when the whole simulation region is very large, some part of the structure is characterized with small geometry feature.

For the above problem, there are basically two solutions. The first one is to use small enough cell size for the whole simulation region. It seems to be good because we can obtain fine results for all the regions. However, after giving it another thought, we may find that this method is not efficient at all. First, small cell size for the whole region means much more calculation time for the simulation. Second, large arrays need to be allocated and requires big memory size. Third, usually this kind of simulation focuses on the region with key elements which are typically small, and there’s no need to make fine grids in the region that can be modeled accurately enough with coarse grid. Last but not
least, there is a limit to decrease the cell size because numerical dispersion and error arise when cell size goes too small.

The second solution would be using a variable lattice of space cells to precisely model the shape and features of a structure. During the initialization steps, automatically generate subgrid at regions that need to be specially treated, while using normal coarse grid for the other parts of the simulating domain. With some extra work to define the non-uniform grids, this method will save much memory space for the program and at the meanwhile keeps the order of accuracy.

### 6.2 Cartesian Subgrids

Along with the advance of FDTD technique, non-uniform meshing for both two dimensional and three dimensional schemes has been paid great attention during the past decades. In [51], one FDTD subgridding algorithm with separated temporal and spatial interfaces is presented. The subgridding has a mesh ratio of 1:3 and was successfully applied in 3D simulation. But no material is allowed to traverse the subgridding interface in that implementation. Paper [52] describes an ADI-FDTD subgridding in ground penetrating radar FDTD models which is based on the unconditionally stable alternating-direction implicit (ADI) FDTD technique [53]. It shows good results with a 1/3 ratio of coarse grid size to fine grid size, but was in 2D implementation. Besides, there are also quasi-nonuniform gridding scheme [54] and non-orthogonal curvilinear coordinate algorithm [55, 56]. In this section, we will investigate the implementation of a 3D
subgridding that can potentially provide high performance with respect to numerical stability and computational efficiency [57].

### 6.2.1 Introduction to the Geometry

The subgridding system has a coarse grid size and sub grid size ratio of 2:1, as depicted in the 2D view Fig.6.1. The two sets of electromagnetic fields for coarse grids and sub grids are denoted in capital and lower case, respectively.

Figure 6.1 Two dimensional view of 2:1 subgridding system
In the subgridding system, the sub grids are embedded inside coarse grid region. The outer boundary of the sub grids doesn’t overlap with the coarse grids’ boundary. Instead, the edge of outer sub grids is 1/4 coarse cell size from the coarse grids. Inside the coarse grids region, the fields updates from the regular FDTD algorithm, while inside the sub grids region, the electromagnetic fields are calculated based on the subgrid update procedure. Between the coarse grids region and sub grids region, it is the intersection layer, as denoted in Fig. 6.1 by the grey region. This layer is used for data exchange, which is crucial for a smoothed calculation. Fig. 6.2 shows the three dimensional view of this 2:1 subgridding system.

Figure 6.2 Three dimensional view of 2:1 subgridding system
6.2.2 Implementation Scheme

As shown in Fig.6.1, there are three calculation regions: coarse grids region, subgrids region and intersection region. For the internal cells, regular FDTD algorithm will be used. For the intersection region, special manipulations are required for the coarse grids region fields and sub grids region fields to interact.

Since the subgrid is nested inside the primary grid, the fields on the boundary of the subgrid cannot be obtained by the normal FDTD algorithm. Those fields need to be calculated by interpolation from the adjoining primary grids. Thus the electromagnetic field is delivered from the primary grid to the subgrid region. That is, when coarse grids fields are passing information to the sub grids fields, spatial interpolation is needed. And temporal interpolation is further needed to obtain sub grids fields at different time steps.

On the other hand, because the subgrid region has a higher accuracy than the primary grid region, after the regular FDTD calculation inside the subgrid region, the fields of the primary grid that is overlapped with subgrid region should be updated based on the interpolation from the subgrid fields. In this way, the high accurate fields from the subgrid region are delivered back into the primary grid region. When sub grids fields are passing information back, spatial interpolation is used to update the coarse grids fields.

Since the sub grids cell size is half of the major grids, the time step $\Delta t$ of sub grids is also half of the coarse grids, which means that each cycles of main grids calculation contains two cycles of sub grids calculation. To conveniently address the update procedure, we denote the electric fields on the outer surfaces of the sub grids regions by $e_x^n$ at time step $n$, and the electric fields of the internal part of the sub grids
regions by $e_{int}^n$. Similarly, we depict the magnetic fields on the outer layer of the sub grids regions by $h_b^n$ at time step $n$, and the magnetic fields of the internal part of the sub grids regions by $h_{int}^n$. In this way, $e_{int}^n$ contains only the fields on the surface. $h_b^n$ contains the fields on the outer surface of sub grids region and the magnetic fields that are half sub-cell away from the surface.

To run one full FDTD cycle for the subgridding system, we assume the following values are already known at the beginning: $E$ and $e$ fields up to time step $n-1$; $H$ fields up to time step $n-0.5$; $h$ fields up to time step $n-0.75$; $e_{int}^n$ fields up to time step $n-0.5$; $h_{int}^n$ fields up to time step $n-0.25$.

<table>
<thead>
<tr>
<th>Time</th>
<th>n-1</th>
<th>n-0.75</th>
<th>n-0.5</th>
<th>n-0.25</th>
<th>n</th>
<th>n+0.25</th>
<th>n+0.5</th>
<th>n+0.75</th>
<th>n+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>$E$</td>
<td></td>
<td></td>
<td></td>
<td>1.</td>
<td>1.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td>$H$</td>
<td></td>
<td>11.</td>
<td>2.</td>
<td>15.</td>
<td>14.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$e$</td>
<td>$e$</td>
<td>$e_{int}$</td>
<td>3.</td>
<td>12.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. $e_s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9$e_{int}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h$</td>
<td>$h$</td>
<td>$h_{int}$</td>
<td>10.</td>
<td>4. $h_b$</td>
<td>13.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.3 Update procedure for 2:1 subgridding system
The update procedure is shown in Fig. 6.3. At the beginning the 6 values in orange are already known and the update is carried out as the following steps: (1) update all E fields from n-1 to n; (2) update all H fields from n-0.5 to n+0.5; (3) update $e_x$ at n based on spatial interpolation using $E^n$; (4) update $h_y$ at n+0.5 based on spatial interpolation using $H^{n+0.5}$; (5) update $e_x$ at n-0.5 based on temporal interpolation using $e_x^{n-2}, e_x^{n-1}$ and $e_x^n$; up to now, all e fields at n-0.5 are known; (6) calculate $h_y$ at n-0.25 based on temporal interpolation using $h_y^{n-1.75}, h_y^{n-0.75}$ and $h_y^{n+0.5}$; (7) calculate $h_y$ at n-0.25 based on Yee algorithm using $e_x^{n-0.5}$; (8) update $h_y$ at n-0.25 by averaging the values from step(6) and step(7) with weighted form $0.35 \times h_y^{n-0.25}_{\text{step 6}} + 0.65 \times h_y^{n-0.25}_{\text{step 7}}$; up to now, all h fields at n-0.25 are known; (9) update $e_{\text{int}}$ at n using Yee algorithm; (10) update all h fields at n+0.25 using Yee algorithm; (11) update $H_{\text{int}}$ which is overlapping with subgrid region at n+0.25, using spatial interpolation; (12) update $e_{\text{int}}$ at n+0.5 using Yee algorithm; (13) update all $h_{\text{int}}$ fields at n+0.75 using Yee algorithm; (14) update $H_{\text{int}}$ at n+0.75, using spatial interpolation; (15) update $H_{\text{int}}$ at n+0.5 by simply averaging results from step(11) and step(14).

The above step (5) and (6) involve temporal interpolation while step (3), (4), (11) and (14) involves spatial interpolation. For temporal interpolation, we use the quadratic function interpolation. For electric fields, we need to obtain $e_x^{n-0.5}$ using $e_x^{n-2}, e_x^{n-1}$ and $e_x^n$, the interpolation equation is given by

$$e_x^{n-0.5} = -0.125e_x^{n-2} + 0.75e_x^{n-1} + 0.375e_x^n \quad (6.1)$$
For magnetic fields, we need to obtain $h_b^{n-0.25}$ using $h_b^{n-1.75}$, $h_b^{n-0.75}$ and $h_b^{n+0.5}$, the interpolation equation is given by

$$h_b^{n-0.25} = -\frac{1}{6} h_b^{n-1.75} + 0.9 h_b^{n-0.75} + \frac{4}{15} h_b^{n+0.5}$$  \hspace{1cm} (6.2)

Interpolation is an important technique here and is directly related to the accuracy of the whole calculation. Common algorithms for interpolation include trilinear interpolation, cubic spline interpolation and cubic smoothing spline interpolation. For simplicity sake, trilinear interpolation is used here. This trilinear algorithm is simple and only involves the cells that next to the one need to update. To reduce the boundary reflection and improve the accuracy, cubic spline or cubic smoothing spline may be utilized [58, 57, 59].

All the above discussions are for dielectric material simulation. When implemented with dispersive material, we have extra terms for electric current density $J$. This term also need to be updated in the sub grids region for every sub grid time step. It will be carried out with the regular ADE update procedure and involves no temporal or spatial interpolations.

### 6.3 Numerical Results

To test our subgridding system, we run the same simulation for Lorentz material as shown in Chapter 2. In this case, a plane wave incidents to a half-space medium characterized by Lorentz material with three pole pairs, as shown in Fig. 6.4. The medium has parameters $\varepsilon_{\infty,p} = 1; \varepsilon_s = 3$ and three Lorentzian resonances in the optical
range: \( f_1 = 2 \times 10^{14} \text{Hz}, \ \delta_1 = 0.5 f_1 \), \( f_2 = 4 \times 10^{14} \text{Hz}, \ \delta_2 = 0.5 f_2 \), \( f_3 = 6 \times 10^{14} \text{Hz}, \ \delta_3 = 0.5 f_3 \). We’ve obtained the reflection coefficient magnitude for three cases. The red curve is the analytical results. The green one is from a simulation with uniform fine grids. And the blue curve comes from a simulation with 2:1 subgridding system, where the subgrid is the same as the green curve simulation.

For the green curve simulation, the structure is 30x20x20 in number of cells in x, y and z directions. The blue curve simulation has the same structure for coarse grid while embedded a 7x7x7 sub grid cube inside the coarse grids region. The comparison with the three methods is shown in Fig.6.5. Again, the figure shows a fairly good match between the three curves. Since we have some reflection from the trilinear interpolation, we expect 2\% error for subgridding system, as can be seen from Fig. 6.5. High order interpolation may be utilized to improve the accuracy.

Figure 6.4 Half space Lorentz material embedded with 2:1 subgridding system
Figure 6.5 Reflection of half space Lorentz material calculated with 2:1 subgridding system

The sub grids space used in this simulation is comparably small. The cell size in x, y and z directions for the uniform grid is 0.01 of wavelength at frequency 1.0e+15Hz. The running time between sub grid simulation and uniform grid simulation are 142 seconds and 255 seconds, respectively, where fairly well acceleration has been achieved.
Chapter 7

Conclusion

7.1 Summary

This dissertation presents the implementation of several key techniques in FDTD algorithm to deal with three dimensional problems characterized with metamaterial structures. We endeavor to make a powerful FDTD engine that aims at high performance when modeling three-dimensional metamaterial structures.

Chapter 1 presents an introduction to the concept and applications of metamaterial, especially when it is integrated with special elements. It also introduces the FDTD technique basics. Chapter 2 talks about the details of FDTD implementation for linear dispersive materials. Then it discusses its utilization in plasmonic array nano-antennas on layered substrates. Chapter 3 emphasizes on the area of nonlinear dispersive materials and gain materials in FDTD. In Chapter 4, the powerful general-purpose analog electronic circuit simulator SPICE is coupled with our FDTD engine, through the implementation of inter-process communication between these two softwares. The work of Chapter 5 and 6 is designed to accelerate the speed of the FDTD simulator. Chapter 5 introduces FDTD the cutting-edge area of GPU computing and targets to make FDTD engine fast and capable to simulate huge structures. Finally, Chapter 6 shows another
method to speed up the FDTD engine when simulating large structures with small-size elements, namely sub-grid meshing.

In conclusion, we successfully implemented several key techniques in FDTD algorithm in order to model three dimensional problems characterized with electromagnetic metamaterials. This implementation method will accelerate metamaterial simulations by providing a promising numerical tool, and hopefully will also make a helpful support for further computational model research.

7.2 Future Research

In Chapter 2 and Chapter 3 we implemented dispersive, nonlinear and gain material model into FDTD algorithm. It can be seen that general procedure may be derived from those cases so that any material could be incorporated into FDTD as long as certain field relation is provided.

In the FDTD-SPICE model, as time steps goes bigger the memory storage in SPICE grows quickly, leading to a decrease in simulation speed. Thus, memory storage clean may be helpful to improve SPICE simulation speed. Also, with multi-SPICE devices simulation, message passing interface (MPI) could be used to make parallel computing for FDTD and SPICE, respectively.

For FDTD-GPU simulation, to overcome the GPU memory limitation of large scale structure simulation with FDTD, divided computational region may be utilized. This is a promising area and will be useful especially for the ever-growing high frequency of the scientific and commercial applications. Suppose we have a bunch of
machines that are connected to the network and equipped with GPUs, if GPU implementation could be integrated with MPI function, further parallelism would make FDTD simulation even faster.

For the implementation of subgridding system, higher order of spatial interpolation is expected to improve the calculation accuracy and reduce the reflection from the coarse grids/sub grids boundary. And higher order of sub grids could also be considered for even smaller structures.
Bibliography


[22] K. J. Blow and D. Wood, "Theoretical description of transient stimulated Raman scattering


[46] Skywork, "SMV1232-079 datasheet".


