

**NUMERICAL SIMULATION OF NANOPULSE
PENETRATION OF BIOLOGICAL MATTER
USING THE ADI-FDTD METHOD**

by

Fei Zhu, B.S., M.S.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

May 2012

UMI Number: 3515668

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

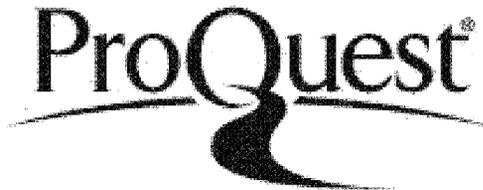
In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3515668

Published by ProQuest LLC 2012. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

03/19/2012

Date

We hereby recommend that the dissertation prepared under our supervision
by Fei Zhu

entitled NUMERICAL SIMULATION OF NANOPULSE PENETRATION OF
BIOLOGICAL MATTERS USING THE ADI-FDTD METHOD

be accepted in partial fulfillment of the requirements for the Degree of
Doctor of Philosophy



Supervisor of Dissertation Research

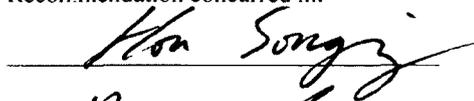


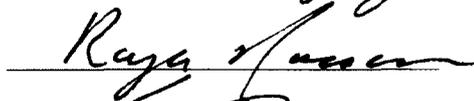
Head of Department

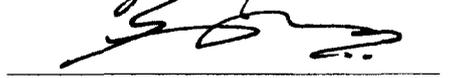
Computational Analysis and Modeling

Department

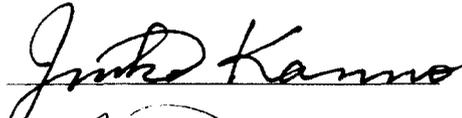
Recommendation concurred in:







Advisory Committee



Approved: 

Director of Graduate Studies

Approved: 

Dean of the Graduate School



Dean of the College

ABSTRACT

Nanopulses are ultra-wide-band (UWB) electromagnetic pulses with pulse duration of only a few nanoseconds and electric field amplitudes greater than 10^5 V/m. They have been widely used in the development of new technologies in the field of medicine. Therefore, the study of the nanopulse bioeffects is important to ensure the appropriate application with nanopulses in biomedical and biotechnological settings. The conventional finite-difference time-domain (FDTD) method for solving Maxwell's equations has been proven to be an effective method to solve the problems related to electromagnetism. However, its application is restricted by the Courant, Friedrichs, and Lewy (CFL) stability condition that confines the time increment and mesh size in the computation in order to prevent the solution from being divergent.

This dissertation develops a new finite difference scheme coupled with the Cole-Cole expression for dielectric coefficients of biological tissues to simulate the electromagnetic fields inside biological tissues when exposed to nanopulses. The scheme is formulated based on the Yee's cell and alternating direction implicit (ADI) technique. The basic idea behind the ADI technique is to break up every time step into two half-time steps. At the first half-step, the finite difference operator on the right-hand side of the Maxwell's equation is implicit only along one coordinate axis direction. At the second half-step, the finite difference operator on the right-hand side of the Maxwell's equation is implicit only along the other coordinate axis direction. As such, only tridiagonal linear

systems are solved. In this numerical method, the Cole-Cole expression is approximated by a second-order Taylor series based on the z -transform method. In addition, the perfectly matched layer is employed for the boundary condition, and the total/scattered field technique is employed to generate the plane wave in order to prevent the wave reflection.

The scheme is tested by numerical examples with two different biological tissues. For the purpose of comparison, both the proposed ADI-FDTD scheme and the conventional FDTD scheme are employed to the numerical examples. The results show that the proposed ADI-FDTD scheme breaks through the CFL stability condition and provides a stable solution with a larger time step, where the conventional FDTD scheme fails. Results also indicate that the computational time can be reduced with a larger time step.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author Fei zhu

Date 3/19/2012

TABLE OF CONTENTS

ABSTRACT.....	iii
LIST OF TABLES.....	viii
LIST OF FIGURES.....	ix
ACKNOWLEDGMENTS.....	xiii
CHAPTER ONE INTRODUCTION.....	1
1.1 General Overview.....	1
1.2 Motivation and Objective of the Research.....	3
1.3 Organization of the Dissertation.....	4
CHAPTER TWO BACKGROUND AND PREVIOUS WORK.....	5
2.1 Electromagnetic Field Equations.....	5
2.1.1 Maxwell's Equations.....	5
2.1.2 Material Properties.....	7
2.1.3 Constitutive Relations.....	9
2.1.4 Maxwell's Equations in a Frequency Dependent Medium.....	10
2.2 FDTD Method.....	11
2.2.1 FDTD Scheme.....	11
2.2.2 Stability Conditions.....	14
2.3 ADI Method.....	15
2.4 Dielectric Properties of Biological Tissues.....	18

2.5 Formulating the Cole-Cole Model	21
2.6 Conclusion.....	24
CHAPTER THREE NUMERICAL SIMULATION.....	26
3.1 Governing Equations for Analyzing Bioeffects	26
3.2 Finite Difference Scheme.....	27
3.3 Boundary Conditions.....	33
3.4 Total/Scattered Field Formulation.....	48
3.5 Source Term	52
3.6 Computational Procedure.....	53
CHAPTER FOUR NUMERICAL EXAMPLES.....	55
4.1 Example Description	55
4.2 Results and Discussions	58
4.2.1 Numerical Verification of the Stability	58
4.2.2 Numerical Accuracy Versus Time Step	75
4.2.3 Comparison of Computational Efficiency.....	87
CHAPTER FIVE CONCLUSION AND FUTURE WORK	89
APPENDIX A NOMENCLATURE.....	91
APPENDIX B SOURCE CODE OF EXAMPLE	93
REFERENCES	120

LIST OF TABLES

Table 4.1 Dielectric properties [20], [48]	57
Table 4.2 Comparison of running time for Case 1	87
Table 4.3 Comparison of running time for Case 2	88

LIST OF FIGURES

Figure 2.1	Positions of the electric and magnetic fields in a Yee's cell	13
Figure 2.2	Transient response of polar dielectric	19
Figure 3.1	Parameters related to the PML	44
Figure 3.2	Contour of E_z from a dipole source in an ADI-FDTD program with a 10-point PML at (a) 65 time steps, (b) 75 time steps, and (c) 85 time steps.....	45
Figure 3.3	Total/Scattered field of the two-dimensional problem space.....	49
Figure 3.4	Total/Scattered field of the three-dimensional problem space.....	50
Figure 4.1	Pulse shape of $E_z = -10^6 \frac{n-100}{\sqrt{200}} e^{-(n-100)^2/400}$ V/m.....	56
Figure 4.2	Simulation results of Case 1 obtained by the conventional FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.....	61
Figure 4.3	Simulation results of Case 1 obtained by the conventional FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 5 time steps, (b) 10 time steps, and (c) 15 time steps.....	62
Figure 4.4	Simulation results of Case 1 obtained by the ADI-FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.....	63
Figure 4.5	Simulation results of Case 1 obtained by the ADI-FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 60 time steps, (b) 72 time steps, (c) 83 time steps, (d) 96 time steps, and (e) 108 time steps.....	64

Figure 4.6	Simulation results of Case 1 obtained by the ADI-FDTD scheme with $4\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 30 time steps, (b) 36 time steps, (c) 42 time steps, (d) 48 time steps, and (e) 54 time steps.....	65
Figure 4.7	Simulation results of Case 1 obtained by the ADI-FDTD scheme with $6\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 20 time steps, (b) 24 time steps, (c) 28 time steps, (d) 32 time steps, and (e) 36 time steps.....	66
Figure 4.8	Simulation results of Case 1 obtained by the ADI-FDTD scheme with $8\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 15 time steps, (b) 18 time steps, (c) 21 time steps, (d) 24 time steps, and (e) 27 time steps.....	67
Figure 4.9	Simulation results of Case 2 obtained by the conventional FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.....	68
Figure 4.10	Simulation results of Case 2 obtained by the conventional FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 5 time steps, (b) 10 time steps, and (c) 15 time steps.....	69
Figure 4.11	Simulation results of Case 2 obtained by the ADI-FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.....	70
Figure 4.12	Simulation results of Ccase 2 obtained by the ADI-FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 60 time steps, (b) 72 time steps, (c) 84 time steps, (d) 96 time steps, and (e) 108 time steps.....	71
Figure 4.13	Simulation results of Case 2 obtained by the ADI-FDTD scheme with $4\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 30 time steps, (b) 36 time steps, (c) 42 time steps, (d) 48 time steps, and (e) 54 time steps.....	72

- Figure 4.14 Simulation results of Case 2 obtained by the ADI-FDTD scheme with $6\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 20 time steps, (b) 24 time steps, (c) 28 time steps, (d) 32 time steps, and (e) 36 time steps.....73
- Figure 4.15 Simulation results of Case 2 obtained by the ADI-FDTD scheme with $8\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 15 time steps, (b) 18 time steps, (c) 21 time steps, (d) 24 time steps, and (e) 27 time steps.....74
- Figure 4.16a Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $120\Delta t_{FDTD}$ 77
- Figure 4.16b Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $144\Delta t_{FDTD}$ 78
- Figure 4.16c Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $168\Delta t_{FDTD}$ 79
- Figure 4.16d Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $192\Delta t_{FDTD}$ 80
- Figure 4.16e Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $216\Delta t_{FDTD}$ 81
- Figure 4.17a Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $120\Delta t_{FDTD}$ 82
- Figure 4.17b Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $144\Delta t_{FDTD}$ 83

- Figure 4.17c Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \text{ }\mu\text{m}$ and $z = 30 \text{ }\mu\text{m}$ after $168\Delta t_{FDTD}$ 84
- Figure 4.17d Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \text{ }\mu\text{m}$ and $z = 30 \text{ }\mu\text{m}$ after $192\Delta t_{FDTD}$ 85
- Figure 4.17e Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \text{ }\mu\text{m}$ and $z = 30 \text{ }\mu\text{m}$ after $216\Delta t_{FDTD}$ 86

ACKNOWLEDGMENTS

Pursuing my Ph.D. degree has been a long journey. Much has happened and changed during this process. Fortunately, my dream has finally come true.

I would like to express my sincere appreciation to my advisor, Dr. Weizhong Dai, for his constant encouragement and patient guidance throughout my five years of study in his lab. He has helped me a lot in improving my research, writing and presentation skills. Without his help and support, this dissertation would not have been completed.

I am also very grateful for having an exceptional doctoral committee. I would like to thank Dr. Raja Nassar, Dr. Jinko Kanno, Dr. Sumeet Dua, and Dr. Songming Hou, for being on my advisory committee for this dissertation and for their valuable advice.

I would also like to thank all my friends. They make my life colorful and full of happiness.

Lastly but most importantly, I would like to thank my family for their love and support. I owe a lot of thanks to my parents, Dijun Zhu and Qindi Yu, for their faith in me and allowing me to chase my dream. I would like to thank my husband, Pan Wang, who accompanies me always on the way to my goal. Their love is always the most powerful support for my life. This dissertation is dedicated to all the above mentioned.

CHAPTER ONE

INTRODUCTION

1.1 General Overview

Nanopulses are ultra-wide-band (UWB) electromagnetic pulses with a pulse duration of only a few nanoseconds and electric field amplitudes greater than 10^5 V/m. They are generated by many electronic devices, including communication instruments such as cell phones. Since the Federal Communication Commission (FCC) adopted the First Report and Order in 2002 [1] that permits the marketing and operation of certain types of new products involving UWB technology, more and more new technologies incorporating nanopulses have been developed or are under consideration. In particular, biomedical technology is one of the important and useful applications of nanopulses, such as medical imaging techniques and electroporation that introduces some substance into a cell. However, inappropriate nanopulse exposure may cause some undesirable health effects, including local temperature effects, interference with the biochemical reactions, alteration of macromolecular structure, and tissue damage [2]. Therefore, it is important to study the bioeffects of nanopulses to ensure the appropriate application of nanopulses in medical settings.

In the past decades, various mathematical models have been developed to investigate the electromagnetic field characteristics inside the tissue exposed to the

nanopulses. Kunz and Luebbers [3] originally developed a finite-difference time-domain (FDTD) program, which solves the Maxwell's equations, to predict the electromagnetic field inside tissues. Based on their FDTD method, Samn and Mathur [4] constructed a mathematical model to estimate the electromagnetic field inside the transverse electromagnetic cell driven by a wideband electromagnetic current pulse. Schoenbach et al. [5] proposed a simple spherical biological cell model to research the relationship between the probability for electric field interactions with cell substructures and the width of electric pulse duration. Joshi et al. [6] carried out a self-consistent model analysis of electroporation in biological matters based on an improved energy model. Joshi and his colleagues [7] also introduced a scheme involving the Laplace, Nernst-Planck, and Smoluchowski equations to study the influence of ultrashort voltage pulses to the temporal dynamics of cell electroporation.

Although the FDTD method [8] has been proven to be an effective method to solve the problems related to the electromagnetism, it is limited by the Courant, Friedrichs, and Lewy (CFL) stability condition [9] for time step. Otherwise, the FDTD method may produce a divergent solution. To date, many techniques have been developed to relax the above restriction. Such techniques include the split-step approach [10], [11] and locally one-dimensional (LOD) FDTD methods [12], [13]. These methods are usually accurate only up to first order in time due to an extra noncommutativity error term. A more efficient method is based on the well-known alternating direction implicit (ADI) technique, which was originally introduced in the mid-1950s for solving parabolic partial differential equations with second order accuracy [14], [15]. In the 1980s, the ADI technique was first applied to Yee's grid for solving the three-dimensional Maxwell's

equations [16]. Based on a new space grid that is different from Yee's Cell, a specially designed two-dimensional FDTD algorithm was presented in [17], which relaxes the CFL condition at the cost of more grid points. Very recently, the unconditionally stable ADI methods were successfully applied to the solution of Maxwell's equations [18], [19].

For many biological tissues, the frequency dependency of dielectric properties is more conveniently given by the Cole-Cole expression [20], [21]. Su et al. [22] has developed an FDTD method coupled with the Cole-Cole expression, where the Cole-Cole expression was approximated by a second-order Taylor series based on the z-transform method [23]. However, the CFL condition is still imposed on this method.

1.2 Motivation and Objective of the Research

In order to investigate the bioeffects of nanopulses, the modeling of biological tissues are necessarily involved in the numerical simulation. The geometry sizes of many biological tissues that are the interest of this study are very small. For example, a normal red cell is only 6-8 μm in diameter. The small size of biological tissue, together with the CFL stability condition, leads to the small space grid and time step that can be employed for the conventional FDTD method. As a result, the number of the iterations needed for simulation may be considerably large for a certain time period, which reduces the computational efficiency.

Considering the drawback of the conventional FDTD method, the objective of this dissertation research is to develop a finite difference time domain scheme without the CFL stability condition. A larger time step used for biological matters will reduce the running time of the program.

1.3 Organization of the Dissertation

Chapter 1 proposes the objective of this dissertation research based on a general literature review.

Chapter 2 introduces the background and previous works related to this research. First of all, the Maxwell's equations for electromagnetic field are discussed in detail. Then, the basic formulations and stability conditions of the conventional FDTD method and ADI scheme are described, respectively. Finally, the dielectric properties of biological tissues and the formulation of the Cole-Cole expression are introduced.

Chapter 3 discusses the numerical simulation in detail. First, governing equations for studying bioeffects are set up. Then, the ADI-FDTD scheme, together with the boundary conditions, the technique for nanopulse simulation, and the pulse source is proposed. The algorithm describing the electromagnetic simulation is presented at the end of the chapter.

Chapter 4 tests the applicability of the proposed scheme by two numerical examples with two different biological tissues, respectively. The stability, numerical accuracy, and computational efficiency are discussed based on the numerical results obtained by both the conventional FDTD method and the proposed ADI-FDTD method.

Chapter 5 gives the conclusion of the research and provides some suggestions for future work.

CHAPTER TWO

BACKGROUND AND PREVIOUS WORK

2.1 Electromagnetic Field Equations

2.1.1 Maxwell's Equations

Maxwell's equations are a set of four partial differential equations that, together with the material dependent constitutive relations and boundary conditions, characterize the fundamental relations between the electric field and the magnetic field. Maxwell's equations were named after the Scottish mathematician and physicist James Clerk Maxwell (1831-1879). They first appeared in the complete form in the “Treatise on Electricity and Magnetism,” which he published in 1873 [24].

The Maxwell's equations can be written in both differential form and integral form. Although these two forms are mathematically equivalent, the differential forms are more convenient for calculating the fields in more complicated situations using numerical methods such as finite difference or finite element methods, while the integral forms are more often used to analytically calculate the fields in simple situations such as the field with symmetric distributions of charges and currents [25].

The general, time-dependent, Maxwell's equations in differential form can be expressed as follows [25]:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad (2.1)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}, \quad (2.2)$$

$$\nabla \cdot \mathbf{D} = \rho, \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (2.4)$$

where the symbols in bold represent vector quantities in three dimensions, and the symbols in italics represent scalar quantities. The quantities involved in Eqs. (2.1) - (2.4) are defined as:

E - the electric field intensity, given in V/m,

H - the magnetic field intensity, given in A/m,

D - the electric flux density, given in C/m²,

B - the magnetic flux density, given in Wb/m²,

J - the current density, given in A/m²,

ρ - the volume electric charge density, given in C/m³.

Equation (2.1) is referred to as Faraday's law of induction. It shows the relationship between a time-varying magnetic field and an induced electric field on the basis of experimental observations made in 1831 by the English scientist Michael Faraday [26]. Faraday's law of induction serves as a basic law of electromagnetism relating to the operating principles of transformers, inductors, and many types of electrical motors and generators [27].

Equation (2.2) is called Ampère's law with Maxwell's correction. Compared with the original Ampère's law, Ampère's law with Maxwell's correction shows that the

magnetic field can be produced by both electrical current (associated with \mathbf{J}) and time-varying electric field (associated with $\partial\mathbf{D}/\partial t$).

Equation (2.3) is known as Gauss's law, which states that the electric flux through any closed surface is proportional to the enclosed electric charge [28]. Although it is more clearly expressed in integral form, the differential form shown in Eq. (2.3) is mathematically equivalent to the integral form due to the divergence theorem.

Equation (2.4) is associated with Gauss's law for magnetism, which shows that the magnetic field has divergence equal to zero. It is equivalent to the statements that there are no magnetic monopoles analogous to electric charges, and that the magnetic flux is conservative.

2.1.2 Material Properties

The electric and magnetic properties of the material are important factors that affect the interaction between the fields and materials. In terms of modeling, material properties define not only what type of computation is needed, but also limit the model [25]. The three basic material properties are conductivity, permittivity, and permeability.

The electrical conductivity is a measure of the ability of a material to conduct an electric current. A high conductivity indicates a material that readily allows the movement of an electric charge. Since the current density \mathbf{J} in a conductor is proportional to the applied electric field intensity \mathbf{E} , the electrical conductivity σ is defined by

$$\sigma = \frac{\mathbf{J}}{\mathbf{E}}. \quad (2.5)$$

The electric properties of dielectric material are largely determined by the polarization of charges within the material due to the presence of an external electric field

[25]. The larger the tendency of electric polarization, the less the electric flux exists in a medium. Permittivity is the measure of this resistance that is encountered when forming an electric field in a medium. Therefore, permittivity is directly related to electric susceptibility, which is a measure of how easily a dielectric polarizes in response to an electric field. Greater electric susceptibility indicates greater permittivity, thereby generating less of the total electric field inside the material. The permittivity is defined by

$$\varepsilon = \varepsilon_0 \varepsilon_r = \varepsilon_0 (1 + \chi_e), \quad (2.6)$$

where ε is the permittivity, ε_0 is the permittivity of free space, ε_r is the relative permittivity of the material, and χ_e is electric susceptibility, which is a dimensionless quantity.

The magnetic properties of materials depend on the interaction between the moving charges within materials and the external magnetic fields. The internal magnetic field is determined by the organization of magnetic dipoles in the medium. Permeability is the measure of the ability of a material to support the formation of a magnetic field within itself. The higher its permeability, the more conductive a material is to a magnetic field. Permeability is commonly defined as

$$\mu = \mu_0 \mu_r, \quad (2.7)$$

Where μ is the permeability, μ_0 is the permeability of free space, and μ_r is the relative permeability.

Technically, conductivity, permittivity, and permeability can all be complex numbers. They are often frequency and temperature dependent.

2.1.3 Constitutive Relations

Maxwell's equations defined by Eqs. (2.1) - (2.4) seems to be equivalent to 12 scalar equations in 12 unknowns of field quantities (\mathbf{E} , \mathbf{H} , \mathbf{D} , and \mathbf{B}). This, however, is not the case. Applying the divergence on both sides of Eq. (2.2) gives

$$\nabla \cdot (\nabla \times \mathbf{H}) = \nabla \cdot \mathbf{J} + \nabla \cdot \frac{\partial \mathbf{D}}{\partial t}. \quad (2.8)$$

Since the equation $\nabla \cdot (\nabla \times \mathbf{A}) = 0$ holds for an arbitrary vector \mathbf{A} , the result obtained is

$$0 = \nabla \cdot \mathbf{J} + \frac{\partial}{\partial t} (\nabla \cdot \mathbf{D}). \quad (2.9)$$

Substituting Eq. (2.3) produces

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t}, \quad (2.10)$$

which is called the electrical continuity equation.

From the above derivation, it can be clearly seen that Eq. (2.3) can be derived from Eq. (2.2) if Eq. (2.10) is postulated. Equation (2.4) can be derived from Eq. (2.1) using the similar derivation. Therefore, only two of the Maxwell's equations, Eq. (2.1) and Eq. (2.2), are independent. In order to apply the Maxwell's equations, two additional relations are added to complete the system [25]:

$$\mathbf{B} = \mu \mathbf{H}, \quad (2.11)$$

$$\mathbf{D} = \varepsilon \mathbf{E}, \quad (2.12)$$

In addition, the current densities \mathbf{J} can be related to the electric field intensity \mathbf{E} with the following equation:

$$\mathbf{J} = \sigma \mathbf{E}. \quad (2.13)$$

Equations (2.11) - (2.13) are referred to as the constitutive relations. They associate the field quantities with the material properties, and determine the important properties of the field equations.

2.1.4 Maxwell's Equations in a Frequency Dependent Medium

For simplicity, it is assumed that the mediums in this research are isotropic, homogeneous, and nonmagnetic: that is, $\mathbf{B} = \mu_0 \mathbf{H}$. Since the conductivity σ and the relative permittivity ϵ_r of most media are frequency-dependent, it is necessary to write down the Maxwell's equations in a more general form including the frequency, as follows [23]:

$$\frac{\partial \mathbf{D}}{\partial t} = \nabla \times \mathbf{H}, \quad (2.14)$$

$$\mathbf{D}(\omega) = \epsilon_0 \cdot \epsilon_r^*(\omega) \cdot \mathbf{E}(\omega), \quad (2.15)$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu_0} \nabla \times \mathbf{E}, \quad (2.16)$$

where $\omega = 2\pi f$ is the angular frequency, f is the ordinary frequency, and ϵ_r^* is the complex relative permittivity including the relative permittivity and conductivity, which can be expressed as follows:

$$\epsilon_r^* = \epsilon_r + \frac{\sigma}{j\omega\epsilon_0}. \quad (2.17)$$

Normalize Eqs. (2.14)-(2.16) with [23]

$$\tilde{E} = \sqrt{\frac{\epsilon_0}{\mu_0}} E, \quad (2.18)$$

and

$$\tilde{\mathbf{D}} = \sqrt{\frac{1}{\epsilon_0 \cdot \mu_0}} \mathbf{D}, \quad (2.19)$$

which leads to

$$\frac{\partial \tilde{\mathbf{D}}}{\partial t} = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \nabla \times \mathbf{H}, \quad (2.20)$$

$$\tilde{\mathbf{D}}(\omega) = \epsilon_r^*(\omega) \cdot \tilde{\mathbf{E}}(\omega), \quad (2.21)$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\sqrt{\epsilon_0 \mu_0}} \nabla \times \tilde{\mathbf{E}}. \quad (2.22)$$

Equations (2.20)-(2.22) are the normalized Maxwell's equations in a frequency dependent medium. \mathbf{D} and \mathbf{E} can still be used instead of $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{E}}$, respectively, in the later discussion in order to simplify the notation.

2.2 FDTD Method

2.2.1 FDTD Scheme

The finite-difference time-domain (FDTD) method, first introduced by Yee in 1966 [29] and later developed by Taflove and others [30]-[35], is one of the important computational electrodynamics modeling techniques. It has been widely used for solving problems related to electromagnetism. The basic theory and applications of the FDTD method are well described and can be found in [9]. Since it is a time-domain method, the FDTD method can cover a wide frequency range with a single simulation run.

For simplicity, time-dependent Maxwell's curl equations in free space are considered, which is written as follows:

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\epsilon_0} \nabla \times \mathbf{H}, \quad (2.23)$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu_0} \nabla \times \mathbf{E}. \quad (2.24)$$

Since \mathbf{E} and \mathbf{H} are vectors in three dimensions, Eqs. (2.23) and (2.24) are equivalent to the following system of scalar equations in the Cartesian coordinates:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon_0} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right), \quad (2.25)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon_0} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right), \quad (2.26)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon_0} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right), \quad (2.27)$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu_0} \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right), \quad (2.28)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu_0} \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right), \quad (2.29)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu_0} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right). \quad (2.30)$$

Yee [29] introduced the following notation for space points and functions of space and time. A grid point in a uniform, rectangular mesh is denoted as

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z). \quad (2.31)$$

And any function F of space and time is denoted as

$$F^n(i, j, k) = F(i\Delta x, j\Delta y, k\Delta z, n\Delta t). \quad (2.32)$$

Here, Δx , Δy , and Δz are, respectively, the space increments in the x , y , and z coordinate directions, and Δt is the time increment, while i , j , k , and n are integers.

In Yee's scheme [29], the computational domain is discretized by using the rectangular grids. For simplicity, the grids are assumed to be same size. Figure 2.1 shows an illustration of a standard Cartesian Yee's cell used for FDTD, and how electric and magnetic field vector components are distributed. It can be seen that each E -field vector component is located midway between a pair of H -field vector components, and conversely. The Yee's scheme can be generalized to variable size and non-orthogonal grid [9].

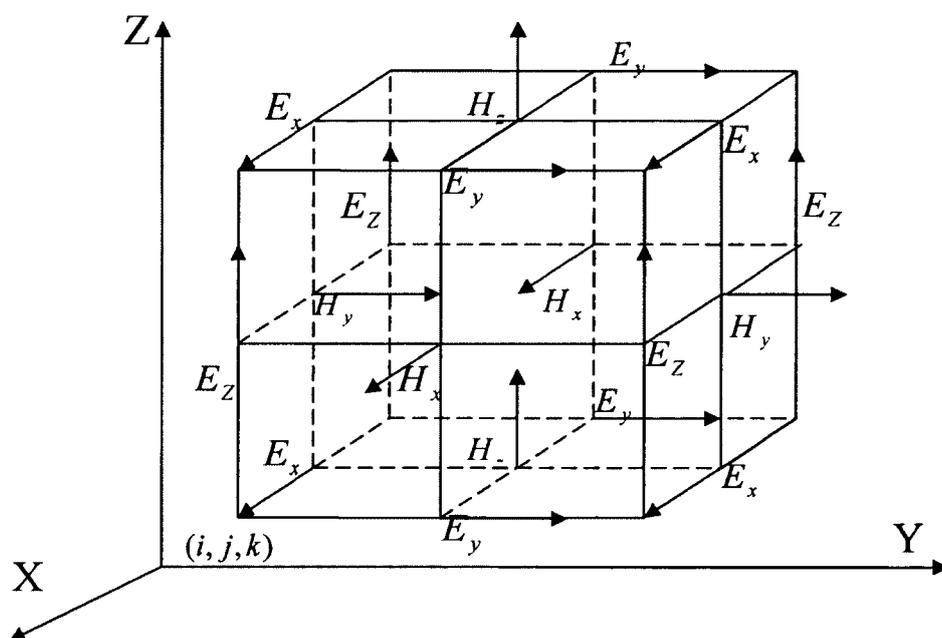


Figure 2.1 Positions of the electric and magnetic fields in a Yee's cell

The FDTD method [29] utilizes the central difference approximation for the space and time derivatives that are second-order accurate in the space and time increments, and then solves the resulting equations numerically to derive the electric and magnetic field

distributions at each time step using an explicit leapfrog scheme. Therefore, Eqs. (2.25) and (2.28) with $\Delta x = \Delta y = \Delta z$ have

$$\begin{aligned}
E_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k) &= E_x^{n-\frac{1}{2}}(i+\frac{1}{2}, j, k) \\
&+ \frac{\Delta t}{\epsilon_0 \Delta x} [H_z^n(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^n(i+\frac{1}{2}, j-\frac{1}{2}, k) \\
&- H_y^n(i+\frac{1}{2}, j, k+\frac{1}{2}) + H_y^n(i+\frac{1}{2}, j, k-\frac{1}{2})], \tag{2.33}
\end{aligned}$$

and

$$\begin{aligned}
H_x^{n+1}(i, j+\frac{1}{2}, k+\frac{1}{2}) &= H_x^n(i, j+\frac{1}{2}, k+\frac{1}{2}) \\
&+ \frac{\Delta t}{\mu_0 \Delta x} [E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+1) - E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k) \\
&- E_z^{n+\frac{1}{2}}(i, j+1, k+\frac{1}{2}) + E_z^{n+\frac{1}{2}}(i, j, k+\frac{1}{2})]. \tag{2.34}
\end{aligned}$$

The FDTD schemes corresponding to Eqs. (2.26)-(2.27) and (2.29)-(2.30), respectively, can be similarly constructed. When Eqs. (2.33) and (2.34) are examined, it can be seen that, at any point in space, the new value of the E -field in time is calculated from the previous value of the E -field and the most recent numerical curl of the local distribution of the H -field in space, and conversely. This is the basic time-stepping relation of the FDTD method [29].

2.2.2 Stability Conditions

The accuracy of the FDTD scheme is first constrained by the cell size. Enough sampling points must be taken to ensure the adequate representation of the computed results. The cell size must be much less than the wavelength at the highest frequency (shortest wavelength) of interest. A good rule of thumb is 10 points per wavelength [23].

Once the cell size is determined, the time increment Δt must be small enough so that it satisfies the following bound, known as the Courant-Friedrichs-Lewy (CFL) stability condition [9], [36]:

$$v\Delta t \leq \left[\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right]^{-1/2}, \quad (2.35)$$

where v is the maximum velocity of propagation in any medium in the program, usually the speed of light in free space. If the time step is not within the bound, the FDTD scheme will become numerically unstable and will produce a divergent solution.

2.3 ADI Method

The alternating direction implicit (ADI) method belongs to a class of techniques called the operator spilling method [37]. It was originally introduced by Peaceman, Douglas and Rachford in the 1950s [38], [39] and later became quite popular in fields such as heat transfer and fluid mechanics. In the 1980's, the ADI scheme was first applied to the Yee's grid for solving the three dimensional Maxwell's equations [16]. However, the scheme suffered instability problems. It was not until 2000 that the three dimensional unconditional stable ADI-FDTD methods were successfully applied to the solution of Maxwell's equations [18], [19].

The basic idea behind the unconditional stable ADI-FDTD method [18] is to break up the time step into two half-steps. And for each half-step, the finite difference operator on the right-hand sides of the Maxwell's equation is implicit only along the single coordinate axis. More specifically, the ADI-FDTD scheme for Eqs. (2.25) and (2.28) with the notations defined by Eqs. (2.31) and (2.32) is as follows:

- 1) For the first half time step (i.e. the advancement from the n th time step to the $(n+1/2)$ th time step)

$$\begin{aligned}
 & \frac{E_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k) - E_x^n(i+\frac{1}{2}, j, k)}{\Delta t/2} \\
 &= \frac{1}{\epsilon_0} \left[\frac{H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j-\frac{1}{2}, k)}{\Delta y} \right. \\
 & \quad \left. - \frac{H_y^n(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^n(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \right], \tag{2.36}
 \end{aligned}$$

and

$$\begin{aligned}
 & \frac{H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+\frac{1}{2}) - H_x^n(i, j+\frac{1}{2}, k+\frac{1}{2})}{\Delta t/2} \\
 &= \frac{1}{\mu_0} \left[\frac{E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+1) - E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k)}{\Delta z} \right. \\
 & \quad \left. - \frac{E_z^n(i, j+1, k+\frac{1}{2}) - E_z^n(i, j, k+\frac{1}{2})}{\Delta y} \right] \tag{2.37}
 \end{aligned}$$

with corresponding expressions for Eqs. (2.26), (2.27), (2.29) and (2.30).

- 2) For the second half time step (i.e. the advancement from the $(n+1/2)$ th time step to the $(n+1)$ th time step)

$$\begin{aligned}
& \frac{E_x^{n+1}(i+\frac{1}{2}, j, k) - E_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k)}{\Delta t/2} \\
&= \frac{1}{\epsilon_0} \left[\frac{H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j-\frac{1}{2}, k)}{\Delta y} \right. \\
&\quad \left. - \frac{H_y^{n+1}(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^{n+1}(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \right], \tag{2.38}
\end{aligned}$$

and

$$\begin{aligned}
& \frac{H_x^{n+1}(i, j+\frac{1}{2}, k+\frac{1}{2}) - H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+\frac{1}{2})}{\Delta t/2} \\
&= \frac{1}{\mu_0} \left[\frac{E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+1) - E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k)}{\Delta z} \right. \\
&\quad \left. - \frac{E_z^{n+1}(i, j+1, k+\frac{1}{2}) - E_z^{n+1}(i, j, k+\frac{1}{2})}{\Delta y} \right] \tag{2.39}
\end{aligned}$$

with corresponding expressions for Eqs. (2.26), (2.27), (2.29) and (2.30).

The ADI-FDTD method displayed above has second-order accuracy in time and space. This system is typically solved using tridiagonal matrix algorithm.

2.4 Dielectric Properties of Biological Tissues

As previously mentioned, the conductivity σ and the relative permittivity ϵ_r of most media vary at different frequencies. The dielectric properties of materials are mainly determined by their complex relative permittivity as shown in Eq. (2.17). The imaginary part of ϵ_r^* represents dielectric losses. Therefore, the dielectric properties of materials are determined as ϵ_r and σ values, as a function of frequency.

The dielectric properties of a biological tissue are due to the interaction of electromagnetic radiation with its constituents at the cellular and molecular level. They are dispersive, and the variation with frequency is very complex [40]. The main features of the dielectric spectrum of tissues are well understood and discussed by Foster and Schwan [41]. Their theoretical analysis is characterized by a single relaxation process centered around a single relaxation time constant. The following description is taken largely from Foster and Schwan [41].

Taking the simplest case as an example, the polarization of a sample will relax towards the steady state as a first order process. The rate at which the response approaches the final value is determined by a single time constant, τ . The transient response is illustrated in Figure 2.2, which has the form:

$$D = D_{\infty} + (D_0 - D_{\infty})(1 - e^{-t/\tau}), \quad (2.40)$$

where D is electric flux density, D_0 is the final value of D , and D_{∞} is the initial value of D . The relaxation time τ is usually determined by the experiments [42].

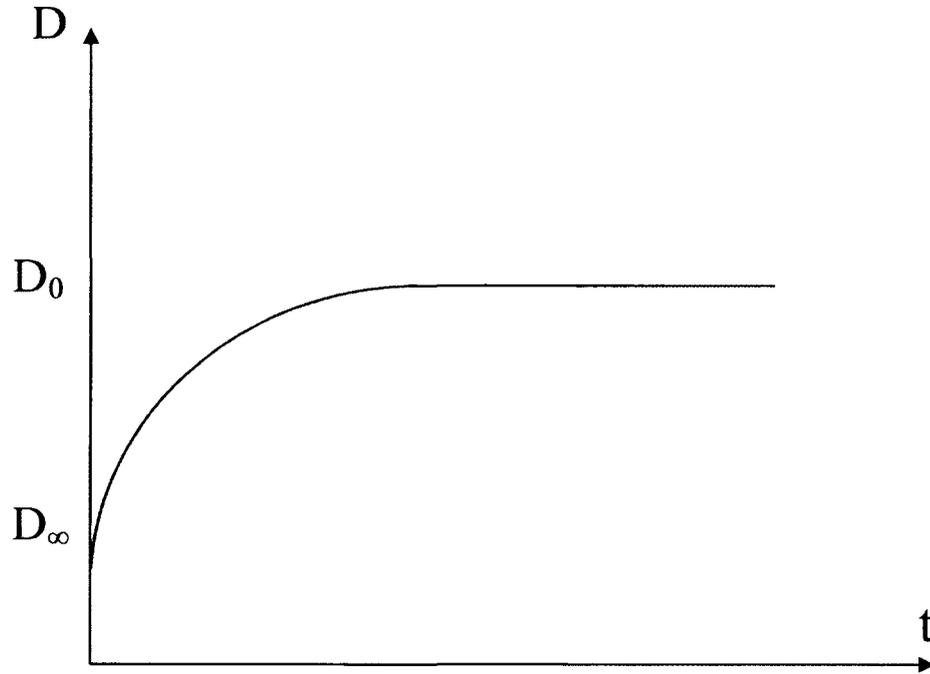


Figure 2.2 Transient response of polar dielectric

Since $D = \epsilon_r^* \epsilon_0 E$, $D_\infty = \epsilon_\infty \epsilon_0 E$, and $D_0 = \epsilon_s \epsilon_0 E$, where ϵ_r^* is the complex relative permittivity, ϵ_∞ is the relative permittivity at infinite frequency ($\omega\tau \gg 1$), at which the polar molecules do not have time to contribute to the polarization [42], and ϵ_s is the static relative permittivity ($\omega\tau \ll 1$), Eq. (2.40) can be rewritten as

$$\epsilon_r^* \epsilon_0 E = \epsilon_\infty \epsilon_0 E + (\epsilon_s \epsilon_0 E - \epsilon_\infty \epsilon_0 E)(1 - e^{-t/\tau}). \quad (2.41)$$

Cancelling $\epsilon_0 E$ terms from both sides and rearranging the equation leaves

$$\epsilon_r^* = \epsilon_\infty + (\epsilon_s - \epsilon_\infty) - (\epsilon_s - \epsilon_\infty)e^{-t/\tau}. \quad (2.42)$$

Taking Laplace transforms on both sides to convert to the frequency domain and simplifying the equation, one obtains

$$\epsilon_r^* = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + s\tau}. \quad (2.43)$$

Setting $s = j\omega$ in order to transform back to the frequency domain gives

$$\varepsilon_r^* = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\omega\tau}, \quad (2.44)$$

which is well known as a single relaxation Debye equation. Debye relaxation is the dielectric relaxation response of an ideal, noninteracting population of dipoles to an alternating external electric field. The magnitude of the dispersion is described as $\Delta\varepsilon = \varepsilon_s - \varepsilon_\infty$. Note that Eq. (2.44) omits the currents flowing at infinite time such as would arise due to the movement of ions in a constant field. Therefore, a static conductivity term, σ_s , is added to the model as follows:

$$\varepsilon_r^* = \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 + j\omega\tau} + \frac{\sigma_s}{j\omega\varepsilon_0}, \quad (2.45)$$

where $\sigma_s = j\omega\varepsilon_0\varepsilon_s$.

The single relaxation model is rare except for some pure polar compounds. For dielectric properties with multiple relaxation times, the complex relative permittivity can be written as [43]:

$$\varepsilon_r^*(\omega) = \varepsilon_\infty + \sum_{m=1}^N \frac{\Delta\varepsilon_m}{1 + j\omega\tau_m} + \frac{\sigma_s}{j\omega\varepsilon_0}, \quad (2.46)$$

where N is the number of relaxation regions.

The above Debye models have often been used to describe the dielectric properties of biological tissues [44]-[46]. However, the structure and composition of biological tissues are both so complicated that each dispersion region may be broadened by multiple contributions to it. The Debye model does not represent the frequency variation of many biological tissues accurately over a wide frequency band. The Cole-Cole expression [47] offers an alternative approach which accounts for the complexity of

the biological tissues by introducing a distribution parameter. The model corresponding to the whole spectrum is written as:

$$\varepsilon_r^*(\omega) = \varepsilon_\infty + \sum_{m=1}^4 \frac{\Delta\varepsilon_m}{1 + (j\omega\tau_m)^{1-\alpha_m}} + \frac{\sigma_s}{j\omega\varepsilon_0}, \quad (2.47)$$

where α ($0 \leq \alpha \leq 1$) is an adjustable parameter that allows for the broadening of the dispersion. This model describes the frequency response of the dielectric properties in the frequency range from Hz to GHz. With a choice of parameters appropriate to each tissue, Eq. (2.47) could be used to predict its dielectric behavior over the desired frequency range. The parameters of the model were adjusted to correspond to a close fit between the model and the most comprehensive data set available for the particular tissue [21], [48].

2.5 Formulating the Cole-Cole Model

As the previous section states, the relative permittivity, $\varepsilon_r^*(\omega)$, appearing in the normalized Maxwell's equation (2.21) is usually described by the Cole-Cole expression as shown in Eq. (2.47). In order to apply the ADI-FDTD method to the solution of the Maxwell's equations expressed by Eqs. (2.20)-(2.22), it needs to convert the Eq. (2.21) together with the Cole-Cole expression from the frequency domain to the time domain. It should be pointed out that it is a challenge to employ the conventional Fourier Transform method for the Cole-Cole expression since the parameters α_m are non-integer. This difficulty is usually solved by reducing the Cole-Cole expression to its predecessors, the Debye model [25], for which $\varepsilon_r^*(\omega)$ is given by

$$\varepsilon_r^*(\omega) = \varepsilon_\infty + \sum_{m=1}^4 \frac{\Delta\tilde{\varepsilon}_m}{1 + (j\omega\tilde{\tau}_m)^1} + \frac{\tilde{\sigma}_s}{j\omega\varepsilon_0}. \quad (2.48)$$

As such, the Debye model is transferred from the frequency domain to the time domain. However, the parameters $\Delta\tilde{\varepsilon}_m$, $\tilde{\tau}_m$, and $\tilde{\sigma}_s$ in Eq. (2.48) are needed to be recalculated since they are different from the parameters $\Delta\varepsilon_m$, τ_m , and σ_s in Eq. (2.47).

In order to use the values of $\Delta\varepsilon_m$, τ_m , and σ_s directly, Dai et al. in 2005 [22] developed a new approach, where the Cole-Cole expression was approximated by a second-order Taylor series based on the z-transform method [23]. The following derivations are taken mainly from previous literatures [49].

Assuming a material is modeled by the Cole-Cole expression, Eq. (2.47), it is substituted into Eq. (2.21), $\mathbf{D}(\omega) = \varepsilon_r^*(\omega) \cdot \mathbf{E}(\omega)$, which gives:

$$\mathbf{D}(\omega) = \varepsilon_\infty \mathbf{E}(\omega) + \sum_{m=1}^4 \frac{\Delta\varepsilon_m \mathbf{E}(\omega)}{1 + (j\omega\tau_m)^{1-\alpha_m}} + \frac{\sigma_s \mathbf{E}(\omega)}{j\omega\varepsilon_0}. \quad (2.49)$$

Introducing

$$\mathbf{I}(\omega) = \frac{\sigma_s \mathbf{E}(\omega)}{j\omega\varepsilon_0}, \quad (2.50)$$

and

$$\mathbf{S}_m(\omega) = \frac{\Delta\varepsilon_m \mathbf{E}(\omega)}{1 + (j\omega\tau_m)^{1-\alpha_m}}, \quad m = 1, 2, 3, 4, \quad (2.51)$$

according to z-transform theorem, by taking the Z transform on Eqs. (2.50) and (2.51), one may obtain

$$\mathbf{I}(z) = \frac{\sigma_s/\varepsilon_0}{1 - z^{-1}} \cdot \mathbf{E}(z) \cdot \Delta t, \quad (2.52)$$

and

$$\mathbf{S}_m(z) = \frac{\Delta \varepsilon_m \mathbf{E}(z)}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m} (1-z^{-1})^{1-\alpha_m}}, \quad m = 1, 2, 3, 4. \quad (2.53)$$

Rearranging the above two equations gives

$$\mathbf{I}(z) = \frac{\sigma_s \Delta t}{\varepsilon_0} \cdot \mathbf{E}(z) + z^{-1} \mathbf{I}(z), \quad (2.54)$$

and

$$\mathbf{S}_m(z) \left[1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m} (1-z^{-1})^{1-\alpha_m} \right] = \Delta \varepsilon_m \mathbf{E}(z), \quad m = 1, 2, 3, 4. \quad (2.55)$$

Employing a second-order Taylor approximation to the term $(1-z^{-1})^{1-\alpha_m}$ as follows:

$$(1-z^{-1})^{1-\alpha_m} \cong 1 - (1-\alpha_m)z^{-1} - \frac{1}{2}(1-\alpha_m)\alpha_m z^{-2}. \quad (2.56)$$

By inserting Eq. (2.56) into Eq. (2.55) and rearranging terms, one obtains

$$\begin{aligned} \mathbf{S}_m(z) &= \frac{\left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}} \left[(1-\alpha_m)z^{-1} \mathbf{S}_m(z) + \frac{1}{2}(1-\alpha_m)\alpha_m z^{-2} \mathbf{S}_m(z) \right] \\ &\quad + \frac{\Delta \varepsilon_m}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}} \mathbf{E}(z). \end{aligned} \quad (2.57)$$

Equation (2.49) in the z-domain then becomes

$$\begin{aligned} \mathbf{D}(z) &= \varepsilon_\infty \mathbf{E}(z) + \sum_{m=1}^4 \mathbf{S}_m(z) + \mathbf{I}(z) \\ &= A \mathbf{E}(z) + \sum_{m=1}^4 B_m \left[(1-\alpha_m)z^{-1} \mathbf{S}_m(z) + \frac{1}{2}(1-\alpha_m)\alpha_m z^{-2} \mathbf{S}_m(z) \right] + z^{-1} \mathbf{I}(z), \end{aligned} \quad (2.58)$$

where

$$A = \varepsilon_\infty + \frac{\sigma_s \Delta t}{\varepsilon_0} + \sum_{m=1}^4 \frac{\Delta \varepsilon_m}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}}, \quad (2.59)$$

$$\begin{aligned} \mathbf{S}_m(z) = B_m \left[(1-\alpha_m) z^{-1} \mathbf{S}_m(z) + \frac{1}{2} (1-\alpha_m) \alpha_m z^{-2} \mathbf{S}_m(z) \right] \\ + \frac{\Delta \varepsilon_m}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}} \mathbf{E}(z), \end{aligned} \quad (2.60)$$

and

$$B_m = \frac{\left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}}, \quad m = 1, 2, 3, 4. \quad (2.61)$$

Thus, Eq. (2.58) can be transferred back to the sampled time domain as follows:

$$\mathbf{E}^n = \frac{1}{A} \left\{ \mathbf{D}^n - \mathbf{I}^{n-1} - \sum_{m=1}^4 B_m \left[(1-\alpha_m) \mathbf{S}_m^{n-1} + \frac{1}{2} (1-\alpha_m) \alpha_m \mathbf{S}_m^{n-2} \right] \right\}, \quad (2.62)$$

where

$$\mathbf{I}^n = \frac{\sigma_s \Delta t}{\varepsilon_0} \cdot \mathbf{E}^n + \mathbf{I}^{n-1}, \quad (2.63)$$

and

$$\mathbf{S}_m^n = B_m \left[(1-\alpha_m) \mathbf{S}_m^{n-1} + \frac{1}{2} (1-\alpha_m) \alpha_m \mathbf{S}_m^{n-2} \right] + \frac{\Delta \varepsilon_m}{1 + \left(\frac{\tau_m}{\Delta t}\right)^{1-\alpha_m}} \mathbf{E}^n. \quad (2.64)$$

2.6 Conclusion

The behaviors of an electromagnetic field are governed by the Maxwell's equations. The dielectric properties of the biological tissue are more conveniently described by the Cole-Cole expression. Since it is difficult to apply the conventional

Fourier Transform method to the Cole-Cole expression to convert it from the frequency domain to the time domain, it is a challenge to solve the Maxwell's equations when coupled to the Cole-Cole expression. In order to overcome this difficulty, a new approach, where the Cole-Cole expression was approximated by a second-order Taylor series based on the z-transform method, was developed.

The FDTD method has been widely used for solving various types of electromagnetic problems. However, the applications of the FDTD method are limited by the CFL stability condition. Very recently, the unconditionally stable ADI-FDTD methods were successfully applied to the solution of Maxwell's equations.

CHAPTER THREE

NUMERICAL SIMULATION

3.1 Governing Equations for Analyzing Bioeffects

For simplicity, it is assumed that the biological tissues being simulated in this research are isotropic, homogeneous, and nonmagnetic, that is, $\mathbf{B} = \mu_0 \mathbf{H}$. Considering the frequency-dependence of the dielectric properties of most biological tissues, the governing equations for studying the nanopulse bioeffects in this dissertation research are the normalized Maxwell's equations as follows [23]:

$$\frac{\partial \tilde{\mathbf{D}}}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \nabla \times \mathbf{H}, \quad (3.1)$$

$$\tilde{\mathbf{D}}(\omega) = \varepsilon_r^*(\omega) \cdot \tilde{\mathbf{E}}(\omega), \quad (3.2)$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\sqrt{\varepsilon_0 \mu_0}} \nabla \times \tilde{\mathbf{E}}, \quad (3.3)$$

where $\tilde{\mathbf{D}}(\omega) = \sqrt{1/(\varepsilon_0 \mu_0)} \mathbf{D}(\omega)$ is the normalized electric flux density, and $\tilde{\mathbf{E}}(\omega) = \sqrt{\varepsilon_0 / \mu_0} \mathbf{E}(\omega)$ is the normalized electric density. In the later sections, the \sim notation will be dropped, and \mathbf{D} and \mathbf{E} will be used instead of $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{E}}$ for simplicity.

In doing three-dimensional simulation, Eqs. (3.1)-(3.3) include 9 different components: E_x , E_y , E_z , D_x , D_y , D_z , H_x , H_y , and H_z . The positions of different

components are illustrated by Yee's cell as shown in Figure 2.1. In addition, since the speed of light $c_0 = 1/\sqrt{\epsilon_0\mu_0}$, the scalar forms of Eqs. (3.1)-(3.3) can be expressed as follows:

$$\frac{\partial D_x}{\partial t} = c_0 \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right), \quad (3.4)$$

$$\frac{\partial D_y}{\partial t} = c_0 \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right), \quad (3.5)$$

$$\frac{\partial D_z}{\partial t} = c_0 \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right), \quad (3.6)$$

$$D_x(\omega) = \epsilon_r^*(\omega) E_x(\omega), \quad (3.7)$$

$$D_y(\omega) = \epsilon_r^*(\omega) E_y(\omega), \quad (3.8)$$

$$D_z(\omega) = \epsilon_r^*(\omega) E_z(\omega), \quad (3.9)$$

$$\frac{\partial H_x}{\partial t} = c_0 \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right), \quad (3.10)$$

$$\frac{\partial H_y}{\partial t} = c_0 \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right), \quad (3.11)$$

$$\frac{\partial H_z}{\partial t} = c_0 \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right). \quad (3.12)$$

3.2 Finite Difference Scheme

It can be seen that the governing Eqs. (3.7)-(3.9) are in the frequency domain, where the relative dielectric constant, $\epsilon_r^*(\omega)$, is modeled by the Cole-Cole expression as shown in Eq. (2.47). Therefore, it needs to convert the Eqs. (3.7)-(3.9) from the frequency domain to the time domain for implementation into the FDTD scheme. Here,

this research follows the method in [22] and employs the z-transform described in [23] and a second-order Taylor approximation proposed in [22]. As a result, \mathbf{E} at time step n can be written as follows:

$$\mathbf{E}^n = \frac{1}{A} \left\{ \mathbf{D}^n - \mathbf{I}^{n-1} - \sum_{m=1}^4 B_m \left[(1-\alpha_m) \mathbf{S}_m^{n-1} + \frac{1}{2} (1-\alpha_m) \alpha_m \mathbf{S}_m^{n-2} \right] \right\}, \quad (3.13)$$

where

$$A = \varepsilon_\infty + \frac{\sigma_s \Delta t}{\varepsilon_0} + \sum_{m=1}^4 \frac{\Delta \varepsilon_m}{1 + \left(\frac{\tau_m}{\Delta t} \right)^{1-\alpha_m}}, \quad (3.14)$$

$$B_m = \frac{\left(\frac{\tau_m}{\Delta t} \right)^{1-\alpha_m}}{1 + \left(\frac{\tau_m}{\Delta t} \right)^{1-\alpha_m}}, \quad m = 1, 2, 3, 4, \quad (3.15)$$

$$\mathbf{I}^n = \frac{\sigma_s \Delta t}{\varepsilon_0} \cdot \mathbf{E}^n + \mathbf{I}^{n-1}, \quad (3.16)$$

$$\mathbf{S}_m^n = B_m \left[(1-\alpha_m) \mathbf{S}_m^{n-1} + \frac{1}{2} (1-\alpha_m) \alpha_m \mathbf{S}_m^{n-2} \right] + \frac{\Delta \varepsilon_m}{1 + \left(\frac{\tau_m}{\Delta t} \right)^{1-\alpha_m}} \mathbf{E}^n, \quad m = 1, 2, 3, 4. \quad (3.17)$$

Equation (3.13) is in the vector form, which has corresponding expressions for E_x , E_y , and E_z , respectively.

To relax the stability constraint of the conventional FDTD method, the ADI-FDTD procedure as described in [18] is used for the governing Eqs. (3.4)-(3.6) and (3.10)-(3.12). The basic idea behind this method is to break up the time step into two half-steps. And for each half-step, the central difference approximations are employed for both the temporal and spatial derivatives in the governing equations, while the finite difference operator on the right-hand sides of the Maxwell's equation is implicit only

along the single coordinate axis. As such, the finite difference scheme for the governing Eqs. (3.4)-(3.6) and (3.10)-(3.12) can be expressed as follows:

- 1) For the first half time step (i.e., the advancement from the n th time step to the $(n+1/2)$ th time step), the first partial derivatives on the right-hand sides of Eqs. (3.4)-(3.6) and (3.10)-(3.12) are taken implicitly, while the second partial derivatives on the right-hand sides of these equations are taken explicitly. The scheme can be written as:

$$D_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k) = D_x^n(i+\frac{1}{2}, j, k) + \frac{c_0 \Delta t}{2} \left[\frac{H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j-\frac{1}{2}, k)}{\Delta y} - \frac{H_y^n(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^n(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \right], \quad (3.18)$$

$$D_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k) = D_y^n(i, j+\frac{1}{2}, k) + \frac{c_0 \Delta t}{2} \left[\frac{H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+\frac{1}{2}) - H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k-\frac{1}{2})}{\Delta z} - \frac{H_z^n(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^n(i-\frac{1}{2}, j+\frac{1}{2}, k)}{\Delta x} \right], \quad (3.19)$$

$$D_z^{n+\frac{1}{2}}(i, j, k+\frac{1}{2}) = D_z^n(i, j, k+\frac{1}{2}) + \frac{c_0 \Delta t}{2} \left[\frac{H_y^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^{n+\frac{1}{2}}(i-\frac{1}{2}, j, k+\frac{1}{2})}{\Delta x} - \frac{H_x^n(i, j+\frac{1}{2}, k+\frac{1}{2}) - H_x^n(i, j-\frac{1}{2}, k+\frac{1}{2})}{\Delta y} \right], \quad (3.20)$$

$$H_x^{n+\frac{1}{2}}(i, j + \frac{1}{2}, k + \frac{1}{2}) = H_x^n(i, j + \frac{1}{2}, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2} \left[\frac{E_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, k + 1) - E_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, k)}{\Delta z} - \frac{E_z^n(i, j + 1, k + \frac{1}{2}) - E_z^n(i, j, k + \frac{1}{2})}{\Delta y} \right], \quad (3.21)$$

$$H_y^{n+\frac{1}{2}}(i + \frac{1}{2}, j, k + \frac{1}{2}) = H_y^n(i + \frac{1}{2}, j, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2} \left[\frac{E_z^{n+\frac{1}{2}}(i + 1, j, k + \frac{1}{2}) - E_z^{n+\frac{1}{2}}(i, j, k + \frac{1}{2})}{\Delta x} - \frac{E_x^n(i + \frac{1}{2}, j, k + 1) - E_x^n(i + \frac{1}{2}, j, k)}{\Delta z} \right], \quad (3.22)$$

$$H_z^{n+\frac{1}{2}}(i + \frac{1}{2}, j + \frac{1}{2}, k) = H_z^n(i + \frac{1}{2}, j + \frac{1}{2}, k) + \frac{c_0 \Delta t}{2} \left[\frac{E_x^{n+\frac{1}{2}}(i + \frac{1}{2}, j + 1, k) - E_x^{n+\frac{1}{2}}(i + \frac{1}{2}, j, k)}{\Delta y} - \frac{E_y^n(i + 1, j + \frac{1}{2}, k) - E_y^n(i, j + \frac{1}{2}, k)}{\Delta x} \right], \quad (3.23)$$

2) For the second half time step (i.e., the advancement from the $(n+1/2)$ th time step to the $(n+1)$ th time step), the first partial derivatives on the right-hand sides of Eqs. (3.4)-(3.6) and (3.10)-(3.12) are taken explicitly, while the second partial derivatives on the right-hand sides of these equations are taken implicitly. The scheme can be written as:

$$D_x^{n+1}(i+\frac{1}{2}, j, k) = D_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k) + \frac{c_0 \Delta t}{2} \left[\frac{H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j-\frac{1}{2}, k)}{\Delta y} - \frac{H_y^{n+1}(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^{n+1}(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \right], \quad (3.24)$$

$$D_y^{n+1}(i, j+\frac{1}{2}, k) = D_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k) + \frac{c_0 \Delta t}{2} \left[\frac{H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+\frac{1}{2}) - H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k-\frac{1}{2})}{\Delta z} - \frac{H_z^{n+1}(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^{n+1}(i-\frac{1}{2}, j+\frac{1}{2}, k)}{\Delta x} \right], \quad (3.25)$$

$$D_z^{n+1}(i, j, k+\frac{1}{2}) = D_z^{n+\frac{1}{2}}(i, j, k+\frac{1}{2}) + \frac{c_0 \Delta t}{2} \left[\frac{H_y^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^{n+\frac{1}{2}}(i-\frac{1}{2}, j, k+\frac{1}{2})}{\Delta x} - \frac{H_x^{n+1}(i, j+\frac{1}{2}, k+\frac{1}{2}) - H_x^{n+1}(i, j-\frac{1}{2}, k+\frac{1}{2})}{\Delta y} \right], \quad (3.26)$$

$$H_x^{n+1}(i, j+\frac{1}{2}, k+\frac{1}{2}) = H_x^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+\frac{1}{2}) + \frac{c_0 \Delta t}{2} \left[\frac{E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k+1) - E_y^{n+\frac{1}{2}}(i, j+\frac{1}{2}, k)}{\Delta z} - \frac{E_z^{n+1}(i, j+1, k+\frac{1}{2}) - E_z^{n+1}(i, j, k+\frac{1}{2})}{\Delta y} \right], \quad (3.27)$$

$$H_y^{n+1}(i+\frac{1}{2}, j, k+\frac{1}{2}) = H_y^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k+\frac{1}{2}) + \frac{c_0 \Delta t}{2} \left[\frac{E_z^{n+\frac{1}{2}}(i+1, j, k+\frac{1}{2}) - E_z^{n+\frac{1}{2}}(i, j, k+\frac{1}{2})}{\Delta x} - \frac{E_x^{n+1}(i+\frac{1}{2}, j, k+1) - E_x^{n+1}(i+\frac{1}{2}, j, k)}{\Delta z} \right], \quad (3.28)$$

$$H_z^{n+1}(i+\frac{1}{2}, j+\frac{1}{2}, k) = H_z^{n+\frac{1}{2}}(i+\frac{1}{2}, j+\frac{1}{2}, k) + \frac{c_0 \Delta t}{2} \left[\frac{E_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j+1, k) - E_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k)}{\Delta y} - \frac{E_y^{n+1}(i+1, j+\frac{1}{2}, k) - E_y^{n+1}(i, j+\frac{1}{2}, k)}{\Delta x} \right], \quad (3.29)$$

Equations (3.18)-(3.29) are a set of implicit equations since the right-hands sides of these equations contain the field values unknown and to be updated. For instance, in Eq. (3.18), one can see that the updated value of D_x is dependent on the stored values of D_x , H_y and unknown values of H_z . By substituting the expressions for $H_z^{n+\frac{1}{2}}$ represented by Eq. (3.23) into Eq. (3.18) and replacing the $E_z^{n+\frac{1}{2}}$ in Eq. (3.23) by Eq. (3.13), a triagonal linear system on $D_x^{n+\frac{1}{2}}$ can be obtained, which can be solved easily by the Thomas algorithm. $D_y^{n+\frac{1}{2}}$, $D_z^{n+\frac{1}{2}}$, D_x^{n+1} , D_y^{n+1} , and D_z^{n+1} can be calculated using similar procedures. Thus, the values of \mathbf{E} and \mathbf{H} fields can be updated by Eqs. (3.13), (3.21)-(3.23), and (3.27)-(3.29).

3.3 Boundary Conditions

Since the computational domain that can be simulated using the ADI-FDTD scheme is limited by computer resources, absorbing boundary conditions (ABCs) play an important role in FDTD simulations for eliminating the unpredictable reflections from the boundary. Otherwise, it is difficult to distinguish the real wave from the reflected junk. There have been a number of techniques on ABCs [3], [9] including the Mur ABC [50], the Liao ABC [51], and various perfectly matched layer (PML) formulations [9], [52]-[54].

The PML method described in [23], which is based on Berenger's PML formulations [52], [53], is employed in this research to implement absorbing boundaries. Different from the conventional boundary conditions, such as Dirichlet or Neumann conditions, the PML method imposes an artificial absorbing layer adjacent to the edges of the computational domain. The wave is attenuated by the absorption and decays exponentially in this absorbing layer.

Suppose a wave is propagating from medium A into medium B, then the amount of reflection at the interface of two media is determined by the reflection coefficient Γ [55]:

$$\Gamma = \frac{\eta_A - \eta_B}{\eta_A + \eta_B}, \quad (3.30)$$

where η_A and η_B are intrinsic impedances of two media, which are determined by permeabilities μ and dielectric constants ε as follows:

$$\eta = \sqrt{\frac{\mu}{\varepsilon}}. \quad (3.31)$$

If η_A equals to η_B , then there is no reflection at the interface of medium A and medium B, and the pulse will continue propagating in medium B. The ideal PML should be a lossy medium that attenuates the wave and makes it die out before it hits the boundary. Therefore, both μ and ε of Eq. (3.31) can be set to be complex numbers considering that the imaginary part of dielectric constant represents the decay.

Following the derivations described in [23] and going back to the governing Eqs. (3.4)-(3.12), they can be written in the Fourier domain as follows:

$$j\omega D_x = c_0 \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right), \quad (3.32)$$

$$j\omega D_y = c_0 \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right), \quad (3.33)$$

$$j\omega D_z = c_0 \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right), \quad (3.34)$$

$$D_x(\omega) = \varepsilon_r^*(\omega) E_x(\omega), \quad (3.35)$$

$$D_y(\omega) = \varepsilon_r^*(\omega) E_y(\omega), \quad (3.36)$$

$$D_z(\omega) = \varepsilon_r^*(\omega) E_z(\omega), \quad (3.37)$$

$$j\omega H_x = c_0 \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right), \quad (3.38)$$

$$j\omega H_y = c_0 \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right), \quad (3.39)$$

$$j\omega H_z = c_0 \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right). \quad (3.40)$$

Here, $c_0 = 1/\sqrt{\epsilon_0\mu_0}$, and $\partial/\partial t$ in Eqs. (3.4)-(3.12) becomes $j\omega$ in Fourier domain in time.

Adding some fictitious dielectric constants and permeabilities into the Eqs. (3.32)-(3.34) and (3.38)-(3.40) to implement the PML gives [56], [57]:

$$j\omega D_x \cdot \epsilon_{fx}^*(x) \cdot \epsilon_{fx}^*(y) \cdot \epsilon_{fx}^*(z) = c_0 \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right), \quad (3.41)$$

$$j\omega D_y \cdot \epsilon_{fy}^*(x) \cdot \epsilon_{fy}^*(y) \cdot \epsilon_{fy}^*(z) = c_0 \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right), \quad (3.42)$$

$$j\omega D_z \cdot \epsilon_{fz}^*(x) \cdot \epsilon_{fz}^*(y) \cdot \epsilon_{fz}^*(z) = c_0 \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right), \quad (3.43)$$

$$j\omega H_x \cdot \mu_{fx}^*(x) \cdot \mu_{fx}^*(y) \cdot \mu_{fx}^*(z) = c_0 \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right), \quad (3.44)$$

$$j\omega H_y \cdot \mu_{fy}^*(x) \cdot \mu_{fy}^*(y) \cdot \mu_{fy}^*(z) = c_0 \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right), \quad (3.45)$$

$$j\omega H_z \cdot \mu_{fz}^*(x) \cdot \mu_{fz}^*(y) \cdot \mu_{fz}^*(z) = c_0 \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right), \quad (3.46)$$

It should be pointed out that the real value of $\epsilon_r^*(\omega)$ in Eqs. (3.35)-(3.37) specifies the medium and it has nothing to do with the fictitious values. Therefore, Eqs. (3.35)-(3.37) keep the same forms in the implementation of PML.

According to the descriptions in [23] and [58], these fictitious values should satisfy two conditions to form a PML:

1) The impedances of background medium (medium A) and PML (medium B) must be a same constant,

$$\eta_A = \eta_B = \sqrt{\frac{\mu_{Fm}^*}{\varepsilon_{Fm}^*}} = 1, \quad m = x, y, \text{ or } z. \quad (3.47)$$

Here, the impedances equal to 1 due to the normalized units.

2) In the direction perpendicular to the boundary, the relative dielectric constant and relative permeability must be the inverse of those in the other directions. For example, in the x direction:

$$\varepsilon_{Fx}^*(x) = \frac{1}{\varepsilon_{Fy}^*(x)} = \frac{1}{\varepsilon_{Fz}^*(x)}, \quad (3.48)$$

$$\mu_{Fx}^*(x) = \frac{1}{\mu_{Fy}^*(x)} = \frac{1}{\mu_{Fz}^*(x)}. \quad (3.49)$$

Thus, the fictitious values are chosen to be complex quantities in the following forms [23]:

$$\varepsilon_{Fx}^*(x) = \mu_{Fx}^*(x) = \left(1 + \frac{\sigma(x)}{j\omega\varepsilon_0}\right)^{-1}, \quad (3.50)$$

$$\varepsilon_{Fy}^*(x) = \varepsilon_{Fz}^*(x) = \mu_{Fy}^*(x) = \mu_{Fz}^*(x) = 1 + \frac{\sigma(x)}{j\omega\varepsilon_0}. \quad (3.51)$$

Obviously, Eqs. (3.50) and (3.51) satisfy two conditions of PML expressed in Eqs. (3.47)-(3.49). If σ increases gradually as it goes into the PML, then D_y , D_z , H_y , and H_z in Eqs. (3.41)-(3.46) will be attenuated.

The fictitious dielectric constants and permeabilities in the y and z directions are obtained using a similar procedure as above. Therefore, the governing Eqs. (3.32)-(3.34) and (3.38)-(3.40) in the PML can be written as:

$$j\omega \cdot \left(1 + \frac{\sigma(x)}{j\omega\varepsilon_0}\right)^{-1} \left(1 + \frac{\sigma(y)}{j\omega\varepsilon_0}\right) \left(1 + \frac{\sigma(z)}{j\omega\varepsilon_0}\right) D_x = c_0 \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right), \quad (3.52)$$

$$j\omega \cdot \left(1 + \frac{\sigma(x)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right)^{-1} \left(1 + \frac{\sigma(z)}{j\omega\epsilon_0}\right) D_y = c_0 \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right), \quad (3.53)$$

$$j\omega \cdot \left(1 + \frac{\sigma(x)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(z)}{j\omega\epsilon_0}\right)^{-1} D_z = c_0 \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right), \quad (3.54)$$

$$j\omega \cdot \left(1 + \frac{\sigma(x)}{j\omega\epsilon_0}\right)^{-1} \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(z)}{j\omega\epsilon_0}\right) H_x = c_0 \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right), \quad (3.55)$$

$$j\omega \cdot \left(1 + \frac{\sigma(x)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right)^{-1} \left(1 + \frac{\sigma(z)}{j\omega\epsilon_0}\right) H_y = c_0 \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right), \quad (3.56)$$

$$j\omega \cdot \left(1 + \frac{\sigma(x)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right) \left(1 + \frac{\sigma(z)}{j\omega\epsilon_0}\right)^{-1} H_z = c_0 \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right). \quad (3.57)$$

To develop the ADI-FDTD formulation, the above equations should be transformed back to the time domain and split into two half-steps in time. To illustrate this idea, Eq. (3.52) is chosen as an example.

For the first half time step, starting by only retaining the dependent fictitious values in x and y directions, Eq. (3.52) is rewritten as

$$j\omega \cdot \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right) D_x = c_0 \left(1 + \frac{\sigma(x)}{j\omega\epsilon_0}\right) \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right). \quad (3.58)$$

Since the left side of Eq. (3.58) can be further expressed as

$$j\omega \cdot \left(1 + \frac{\sigma(y)}{j\omega\epsilon_0}\right) D_x = j\omega D_x + \frac{\sigma(y)}{\epsilon_0} D_x, \quad (3.59)$$

transforming it to the time domain and then taking the finite difference approximations give

$$\begin{aligned} \frac{\partial D_x}{\partial t} + \frac{\sigma(y)}{\varepsilon_0} D_x &\cong \frac{D_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) - D_x^n\left(i+\frac{1}{2}, j, k\right)}{\Delta t/2} \\ &+ \frac{\sigma(j)}{\varepsilon_0} \frac{D_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) + D_x^n\left(i+\frac{1}{2}, j, k\right)}{2}, \end{aligned} \quad (3.60)$$

From the right-hand side of Eq. (3.58), one may use the fact that $(1/j\omega)$ can be regarded as an integration operator over time:

$$\begin{aligned} &c_0 \left(1 + \frac{\sigma(x)}{j\omega\varepsilon_0}\right) \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) \\ &= c_0 \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) + c_0 \frac{\sigma(x)}{j\omega\varepsilon_0} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right) \\ &= c_0 \left[\frac{H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)}{\Delta y} - \frac{H_y^n\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) - H_y^n\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)}{\Delta z} \right] \\ &+ c_0 \frac{\sigma\left(i+\frac{1}{2}\right)}{\varepsilon_0} \left[\frac{\Delta t}{2} \frac{H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)}{\Delta y} \right. \\ &\left. + \frac{\Delta t}{2} \sum_{l=0}^n \text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right) + \frac{\Delta t}{2} \sum_{l=0}^{n-1} \text{curl}_- H_x^{l+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) \right], \end{aligned} \quad (3.61)$$

where

$$\begin{aligned} \text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right) &= \frac{H_z^l\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^l\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)}{\Delta y} \\ &- \frac{H_y^l\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) - H_y^l\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)}{\Delta z}. \end{aligned} \quad (3.62)$$

Equations (3.60) and (3.61) are now substituted into Eq. (3.58). This gives

$$\begin{aligned}
D_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) &= g_{j3}(j) D_x^n\left(i+\frac{1}{2}, j, k\right) \\
&+ \frac{c_0 \Delta t}{2 \Delta y} g_{j2}(j) \left(1+g_{i1}\left(i+\frac{1}{2}\right)\right) \left[H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right)-H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)\right] \\
&- \frac{c_0 \Delta t}{2 \Delta z} g_{j2}(j) \left[H_y^n\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right)-H_y^n\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)\right] \\
&+ \frac{c_0 \Delta t}{2} g_{j2}(j) g_{i1}\left(i+\frac{1}{2}\right) \left[\sum_{l=0}^n \text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right)+\sum_{l=0}^{n-1} \text{curl}_- H_x^{l+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right)\right]. \quad (3.63)
\end{aligned}$$

The parameters g_{j3} , g_{j2} , and g_{i1} are defined as

$$g_{j3}(j) = \frac{1-\sigma(j)\Delta t/4\epsilon_0}{1+\sigma(j)\Delta t/4\epsilon_0}, \quad (3.64)$$

$$g_{j2}(j) = \frac{1}{1+\sigma(j)\Delta t/4\epsilon_0}, \quad (3.65)$$

$$g_{i1}(i) = \frac{\sigma(i)\Delta t}{2\epsilon_0}. \quad (3.66)$$

Equation (3.63) is only the ADI-FDTD formulation for Eq. (3.58), where the z dependent term from Eq. (3.52) was set aside. Using the same procedure as above, Eq. (3.52) can be approximated by

$$\begin{aligned}
D_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) &= g_{j3}(j) g_{k3}(k) D_x^n\left(i+\frac{1}{2}, j, k\right) \\
&+ \frac{c_0 \Delta t}{2 \Delta y} g_{j2}(j) g_{k2}(k) \left(1+g_{i1}\left(i+\frac{1}{2}\right)\right) \left[H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right)-H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)\right] \\
&- \frac{c_0 \Delta t}{2 \Delta z} g_{j2}(j) g_{k2}(k) \left[H_y^n\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right)-H_y^n\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)\right] \\
&+ \frac{c_0 \Delta t}{2} g_{j2}(j) g_{k2}(k) g_{i1}\left(i+\frac{1}{2}\right) \\
&\cdot \left[\sum_{l=0}^n \text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right)+\sum_{l=0}^{n-1} \text{curl}_- H_x^{l+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right)\right], \quad (3.67)
\end{aligned}$$

where

$$g_{k3}(k) = \frac{1 - \sigma(k)\Delta t/4\epsilon_0}{1 + \sigma(k)\Delta t/4\epsilon_0}, \quad (3.68)$$

$$g_{k2}(k) = \frac{1}{1 + \sigma(k)\Delta t/4\epsilon_0}. \quad (3.69)$$

So far, the ADI-FDTD scheme has been obtained for the governing Eq. (3.52).

Following the same derivation as for Eq. (3.52), the full ADI-FDTD scheme for governing Eqs. (3.52)-(3.57) including PML can be expressed as follows:

1) For the first half time step (i.e., at the $(n+1/2)$ th time step),

$$\begin{aligned} D_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) &= g_{j3}(j) g_{k3}(k) D_x^n\left(i+\frac{1}{2}, j, k\right) \\ &+ \frac{c_0\Delta t}{2\Delta y} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1}\left(i+\frac{1}{2}\right)\right) \left[H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right) \right] \\ &- \frac{c_0\Delta t}{2\Delta z} g_{j2}(j) g_{k2}(k) \left[H_y^n\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) - H_y^n\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right) \right] \\ &+ \frac{c_0\Delta t}{2} g_{j2}(j) g_{k2}(k) g_{i1}\left(i+\frac{1}{2}\right) \\ &\cdot \left[\sum_{l=0}^n \text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right) + \sum_{l=0}^{n-1} \text{curl}_- H_x^{l+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) \right] \end{aligned} \quad (3.70)$$

and similar corresponding expressions for $D_y^{n+\frac{1}{2}}\left(i, j+\frac{1}{2}, k\right)$ and $D_z^{n+\frac{1}{2}}\left(i, j, k+\frac{1}{2}\right)$.

$$\begin{aligned}
H_x^{n+\frac{1}{2}}\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) &= f_{j3}\left(j+\frac{1}{2}\right) f_{k3}\left(k+\frac{1}{2}\right) H_x^n\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) \\
&+ \frac{c_0 \Delta t}{2 \Delta z} f_{j2}\left(j+\frac{1}{2}\right) f_{k2}\left(k+\frac{1}{2}\right) \left(1+f_{i1}(i)\right) \left[E_y^{n+\frac{1}{2}}\left(i, j+\frac{1}{2}, k+1\right)-E_y^{n+\frac{1}{2}}\left(i, j+\frac{1}{2}, k\right)\right] \\
&- \frac{c_0 \Delta t}{2 \Delta y} f_{j2}\left(j+\frac{1}{2}\right) f_{k2}\left(k+\frac{1}{2}\right) \left[E_z^n\left(i, j+1, k+\frac{1}{2}\right)-E_z^n\left(i, j, k+\frac{1}{2}\right)\right] \\
&+ \frac{c_0 \Delta t}{2} f_{j2}\left(j+\frac{1}{2}\right) f_{k2}\left(k+\frac{1}{2}\right) f_{i1}(i) \\
&\cdot \left[\sum_{l=0}^n \text{curl}_- E_x^l\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right)+\sum_{l=0}^{n-1} \text{curl}_- E_x^{l+\frac{1}{2}}\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right)\right]
\end{aligned} \tag{3.71}$$

and similar corresponding expressions for $H_y^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right)$ and $H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right)$.

2) For the second half time step (i.e., at the $(n+1)$ th time step),

$$\begin{aligned}
D_x^{n+1}\left(i+\frac{1}{2}, j, k\right) &= g_{j3}(j) g_{k3}(k) D_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right) \\
&+ \frac{c_0 \Delta t}{2 \Delta y} g_{j2}(j) g_{k2}(k) \left[H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right)-H_z^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)\right] \\
&- \frac{c_0 \Delta t}{2 \Delta z} g_{j2}(j) g_{k2}(k) \left(1+g_{i1}\left(i+\frac{1}{2}\right)\right) \left[H_y^{n+1}\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right)-H_y^{n+1}\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)\right] \\
&+ \frac{c_0 \Delta t}{2} g_{j2}(j) g_{k2}(k) g_{i1}\left(i+\frac{1}{2}\right) \\
&\cdot \left[\sum_{l=0}^n \text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right)+\sum_{l=0}^n \text{curl}_- H_x^{l+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right)\right]
\end{aligned} \tag{3.72}$$

and similar corresponding expressions for $D_y^{n+1}\left(i, j+\frac{1}{2}, k\right)$ and $D_z^{n+1}\left(i, j, k+\frac{1}{2}\right)$.

$$\begin{aligned}
H_x^{n+1}\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) &= f_{j3}\left(j+\frac{1}{2}\right) f_{k3}\left(k+\frac{1}{2}\right) H_x^n\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) \\
&+ \frac{c_0 \Delta t}{2 \Delta z} f_{j2}\left(j+\frac{1}{2}\right) f_{k2}\left(k+\frac{1}{2}\right) \left[E_y^{n+\frac{1}{2}}\left(i, j+\frac{1}{2}, k+1\right) - E_y^{n+\frac{1}{2}}\left(i, j+\frac{1}{2}, k\right) \right] \\
&- \frac{c_0 \Delta t}{2 \Delta y} f_{j2}\left(j+\frac{1}{2}\right) f_{k2}\left(k+\frac{1}{2}\right) (1+f_{i1}(i)) \left[E_z^{n+1}\left(i, j+1, k+\frac{1}{2}\right) - E_z^{n+1}\left(i, j, k+\frac{1}{2}\right) \right] \\
&+ \frac{c_0 \Delta t}{2} f_{j2}\left(j+\frac{1}{2}\right) f_{k2}\left(k+\frac{1}{2}\right) f_{i1}(i) \\
&\cdot \left[\sum_{l=0}^n \text{curl}_- E_x^l\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) + \sum_{l=0}^n \text{curl}_- E_x^{l+\frac{1}{2}}\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) \right]
\end{aligned}$$

(3.73)

and similar corresponding expressions for $H_y^{n+1}\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right)$ and $H_z^{n+1}\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right)$.

Here,

$$g_{i1}(i) = \frac{\sigma(i) \Delta t}{2 \varepsilon_0}, \quad g_{i2}(i) = \frac{1}{1 + \sigma(i) \Delta t / 4 \varepsilon_0}, \quad g_{i3}(i) = \frac{1 - \sigma(i) \Delta t / 4 \varepsilon_0}{1 + \sigma(i) \Delta t / 4 \varepsilon_0}, \quad (3.74)$$

$$f_{i1}(i) = \frac{\sigma(i) \Delta t}{2 \varepsilon_0}, \quad f_{i2}(i) = \frac{1}{1 + \sigma(i) \Delta t / 4 \varepsilon_0}, \quad f_{i3}(i) = \frac{1 - \sigma(i) \Delta t / 4 \varepsilon_0}{1 + \sigma(i) \Delta t / 4 \varepsilon_0} \quad (3.75)$$

and similar corresponding expressions for j and k , as well as

$$\begin{aligned}
\text{curl}_- H_x^l\left(i+\frac{1}{2}, j, k\right) &= \frac{H_z^l\left(i+\frac{1}{2}, j+\frac{1}{2}, k\right) - H_z^l\left(i+\frac{1}{2}, j-\frac{1}{2}, k\right)}{\Delta y} \\
&- \frac{H_y^l\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) - H_y^l\left(i+\frac{1}{2}, j, k-\frac{1}{2}\right)}{\Delta z}, \quad (3.76)
\end{aligned}$$

$$\begin{aligned}
\text{curl}_- E_x^l\left(i, j+\frac{1}{2}, k+\frac{1}{2}\right) &= \frac{E_y^l\left(i, j+\frac{1}{2}, k+1\right) - E_y^l\left(i, j+\frac{1}{2}, k\right)}{\Delta z} \\
&- \frac{E_z^l\left(i, j+1, k+\frac{1}{2}\right) - E_z^l\left(i, j, k+\frac{1}{2}\right)}{\Delta y} \quad (3.77)
\end{aligned}$$

together with similar corresponding expressions for the y and z directions.

As suggested in [23], an auxiliary parameter is employed instead of actually varying the conductivities in calculating the parameters g and f expressed by Eqs. (3.74) and (3.75):

$$x_n = \frac{\sigma \Delta t}{4\epsilon_0}, \quad (3.78)$$

which increases as it goes into the PML. Here, an empirical formula given in [23] is employed in the computation,

$$x_n(i) = 0.333 \cdot \left(\frac{i}{\text{length_pml}} \right)^3, \quad i = 1, 2, \dots, \text{length_pml}, \quad (3.79)$$

where length_pml is the length of PML. Thus, parameters g and f can be calculated as follows:

$$g_{i1}(i) = 2x_n(i), \quad g_{i2}(i) = \frac{1}{1+x_n(i)}, \quad g_{i3}(i) = \frac{1-x_n(i)}{1+x_n(i)}, \quad (3.80)$$

$$f_{i1}(i) = 2x_n(i), \quad f_{i2}(i) = \frac{1}{1+x_n(i)}, \quad f_{i3}(i) = \frac{1-x_n(i)}{1+x_n(i)} \quad (3.81)$$

and similar corresponding expressions for j and k . It should be pointed out that f_{i1} , g_{i2} , and g_{i3} are computed at the full intervals, i ; while g_{i1} , f_{i2} , and f_{i3} are computed at the half intervals, $i + 1/2$. On the main problem space, g and f are defined as

$$g_{i1}(i) = 0, \quad g_{i2}(i) = g_{i3}(i) = 1, \quad (3.82)$$

$$f_{i1}(i) = 0, \quad f_{i2}(i) = f_{i3}(i) = 1 \quad (3.83)$$

and similar definitions for j and k . Therefore, the above parameters create a seamless transition from the main problem space to the PML, as shown in Figure 3.1.

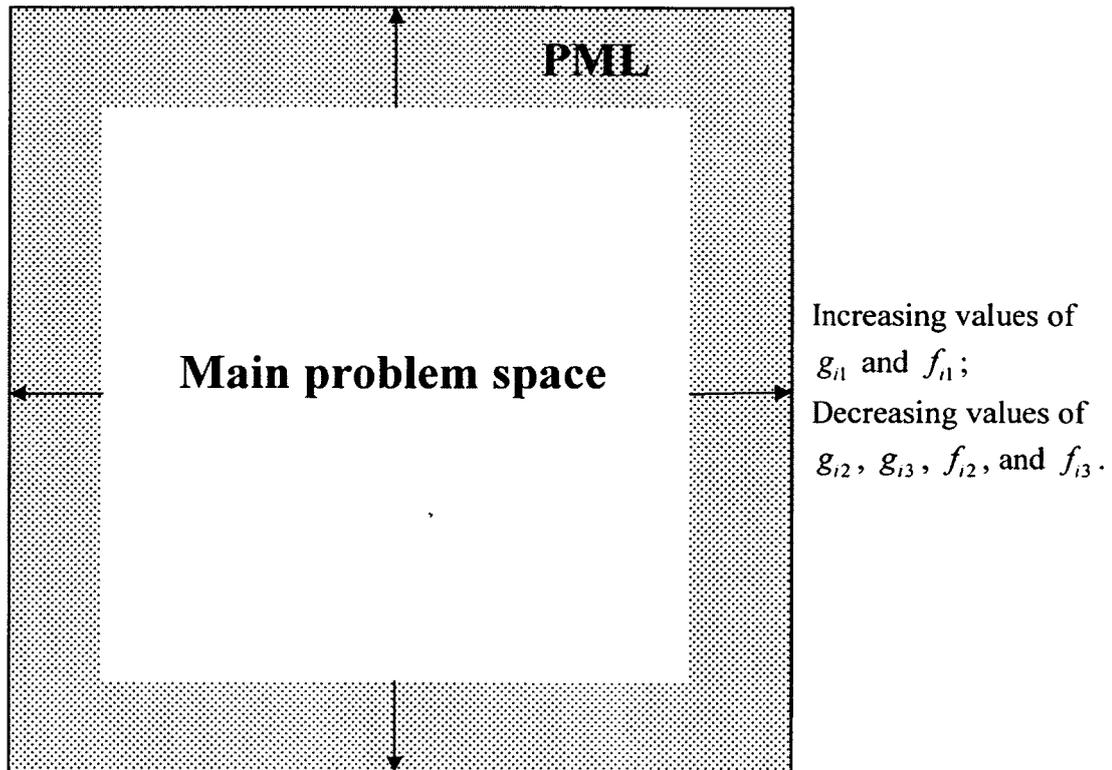
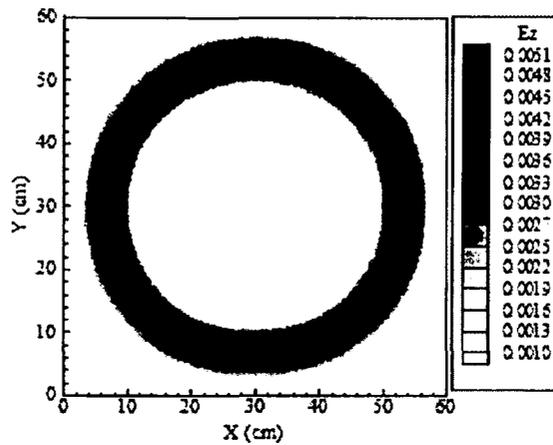
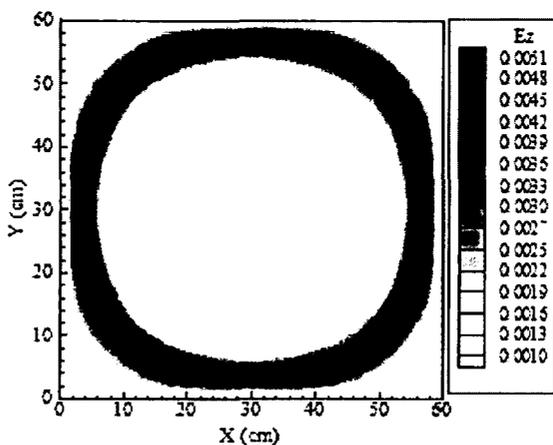


Figure 3.1 Parameters related to the PML

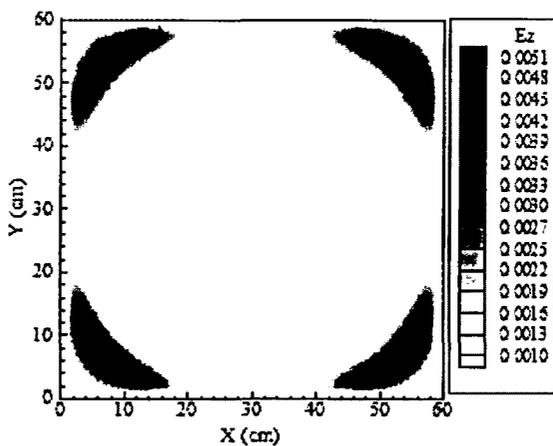
Figure 3.2 illustrates the effectiveness of a 10-point PML of a three-dimensional simulation in free space. A dipole source is located at the center of the problem space. It shows the contours of E_z in the xy cross-section. It can be seen from this figure that the wave is absorbed as it reaches the PML, and that no reflection occurs.



(a)



(b)



(c)

Figure 3.2 Contour of E_z from a dipole source in an ADI-FDTD program with a 10-point PML at (a) 65 time steps, (b) 75 time steps, and (c) 85 time steps.

It should be pointed out that Eqs. (3.70)-(3.73) cannot be used for direct numerical calculation since they include unknown field components on both sides. By substituting the corresponding expression for $H_z^{n+\frac{1}{2}}$ similar to Eq. (3.71) and $E_x^{n+\frac{1}{2}}$ represented by Eq. (3.13) into Eq. (3.70), one may obtain

$$-P \cdot D_x^{n+\frac{1}{2}} \left(i + \frac{1}{2}, j-1, k \right) + Q \cdot D_x^{n+\frac{1}{2}} \left(i + \frac{1}{2}, j, k \right) - R \cdot D_x^{n+\frac{1}{2}} \left(i + \frac{1}{2}, j+1, k \right) = W, \quad (3.84)$$

where

$$P = \frac{c_0^2 \Delta t^2}{4 \Delta y^2} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) f_{i2} \left(i + \frac{1}{2} \right) f_{j2} \left(j - \frac{1}{2} \right) (1 + f_{k1}(k)) \frac{1}{A}, \quad (3.85)$$

$$Q = 1 + \frac{c_0^2 \Delta t^2}{4 \Delta y^2} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) \cdot f_{i2} \left(i + \frac{1}{2} \right) \left(f_{j2} \left(j - \frac{1}{2} \right) + f_{j2} \left(j + \frac{1}{2} \right) \right) (1 + f_{k1}(k)) \frac{1}{A}, \quad (3.86)$$

$$R = \frac{c_0^2 \Delta t^2}{4 \Delta y^2} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) f_{i2} \left(i + \frac{1}{2} \right) f_{j2} \left(j + \frac{1}{2} \right) (1 + f_{k1}(k)) \frac{1}{A}, \quad (3.87)$$

$$\begin{aligned}
W = & g_{j3}(j) g_{k3}(k) D_x^n \left(i + \frac{1}{2}, j, k \right) \\
& - \frac{c_0 \Delta t}{2 \Delta z} g_{j2}(j) g_{k2}(k) \left[H_y^n \left(i + \frac{1}{2}, j, k + \frac{1}{2} \right) - H_y^n \left(i + \frac{1}{2}, j, k - \frac{1}{2} \right) \right] \\
& + \frac{c_0 \Delta t}{2} g_{j2}(j) g_{k2}(k) g_{i1} \left(i + \frac{1}{2} \right) \left[\sum_{l=0}^n \text{curl}_- H_x^l \left(i + \frac{1}{2}, j, k \right) + \sum_{l=0}^{n-1} \text{curl}_- H_x^{l+\frac{1}{2}} \left(i + \frac{1}{2}, j, k \right) \right] \\
& + \frac{c_0 \Delta t}{2 \Delta y} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) \left\{ f_{j3} \left(i + \frac{1}{2} \right) f_{j3} \left(j + \frac{1}{2} \right) H_z^n \left(i + \frac{1}{2}, j + \frac{1}{2}, k \right) \right. \\
& - \frac{c_0 \Delta t}{2 \Delta x} f_{j2} \left(i + \frac{1}{2} \right) f_{j2} \left(j + \frac{1}{2} \right) \left[E_y^n \left(i + 1, j + \frac{1}{2}, k \right) - E_y^n \left(i, j + \frac{1}{2}, k \right) \right] \\
& + \frac{c_0 \Delta t}{2} f_{j2} \left(i + \frac{1}{2} \right) f_{j2} \left(j + \frac{1}{2} \right) f_{k1}(k) \\
& \cdot \left[\sum_{l=0}^n \text{curl}_- E_z^l \left(i + \frac{1}{2}, j + \frac{1}{2}, k \right) + \sum_{l=0}^{n-1} \text{curl}_- E_z^{l+\frac{1}{2}} \left(i + \frac{1}{2}, j + \frac{1}{2}, k \right) \right] \\
& - f_{i3} \left(i + \frac{1}{2} \right) f_{j3} \left(j - \frac{1}{2} \right) H_z^n \left(i + \frac{1}{2}, j - \frac{1}{2}, k \right) \\
& + \frac{c_0 \Delta t}{2 \Delta x} f_{j2} \left(i + \frac{1}{2} \right) f_{j2} \left(j - \frac{1}{2} \right) \left[E_y^n \left(i + 1, j - \frac{1}{2}, k \right) - E_y^n \left(i, j - \frac{1}{2}, k \right) \right] \\
& - \frac{c_0 \Delta t}{2} f_{j2} \left(i + \frac{1}{2} \right) f_{j2} \left(j - \frac{1}{2} \right) f_{k1}(k) \\
& \cdot \left[\sum_{l=0}^n \text{curl}_- E_z^l \left(i + \frac{1}{2}, j - \frac{1}{2}, k \right) + \sum_{l=0}^{n-1} \text{curl}_- E_z^{l+\frac{1}{2}} \left(i + \frac{1}{2}, j - \frac{1}{2}, k \right) \right] \Big\} \\
& + \frac{c_0^2 \Delta t^2}{4 \Delta y^2} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) f_{j2} \left(i + \frac{1}{2} \right) f_{j2} \left(j - \frac{1}{2} \right) (1 + f_{k1}(k)) \frac{1}{A} \\
& \cdot \left\{ -I_x^n \left(i + \frac{1}{2}, j - 1, k \right) - \sum_{m=1}^4 B_m \left[(1 - \alpha_m) S_m^n + \frac{(1 - \alpha_m) \alpha_m}{2} S_m^{n-\frac{1}{2}} \right] \left(i + \frac{1}{2}, j - 1, k \right) \right\} \\
& - \frac{c_0^2 \Delta t^2}{4 \Delta y^2} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) f_{j2} \left(i + \frac{1}{2} \right) \left(f_{j2} \left(j - \frac{1}{2} \right) + f_{j2} \left(j + \frac{1}{2} \right) \right) (1 + f_{k1}(k)) \frac{1}{A} \\
& \cdot \left\{ -I_x^n \left(i + \frac{1}{2}, j, k \right) - \sum_{m=1}^4 B_m \left[(1 - \alpha_m) S_m^n + \frac{(1 - \alpha_m) \alpha_m}{2} S_m^{n-\frac{1}{2}} \right] \left(i + \frac{1}{2}, j, k \right) \right\} \\
& + \frac{c_0^2 \Delta t^2}{4 \Delta y^2} g_{j2}(j) g_{k2}(k) \left(1 + g_{i1} \left(i + \frac{1}{2} \right) \right) f_{j2} \left(i + \frac{1}{2} \right) f_{j2} \left(j + \frac{1}{2} \right) (1 + f_{k1}(k)) \frac{1}{A} \\
& \cdot \left\{ -I_x^n \left(i + \frac{1}{2}, j + 1, k \right) - \sum_{m=1}^4 B_m \left[(1 - \alpha_m) S_m^n + \frac{(1 - \alpha_m) \alpha_m}{2} S_m^{n-\frac{1}{2}} \right] \left(i + \frac{1}{2}, j + 1, k \right) \right\}, \quad (3.88)
\end{aligned}$$

Equation (3.88) provides the implicit update expression for D_x at the $(n+1/2)$ time step, which is a tridiagonal linear system with respect to index j . This can be solved easily by the Thomas algorithm. Using similar procedures, $D_y^{n+\frac{1}{2}}$, $D_z^{n+\frac{1}{2}}$, D_x^{n+1} , D_y^{n+1} , and D_z^{n+1} can be obtained.

3.4 Total/Scattered Field Formulation

It is necessary to employ a plane-wave simulation for many practical problems in computational electromagnetics [3], [9]. The simulation of plane wave relies on the total/scattered field formulation, which introduces a plane wave boundary that generates the wave at one side and subtracts out the wave at the other side [59], [60]. The reason for doing this is to minimize the interaction between the propagating plane wave and the perfectly matched layer, and then to minimize the computational load on the perfectly matched layer [23].

In order to implement the total/scattered formulation, the problem space is divided into two regions: the total field and the scattered field. Figure 3.3 illustrates how this is accomplished in a two-dimensional problem space. For the case shown in Figure 3.3, the incident plane wave is generated at $j = ja$ and subtracted out at $j = jb$. The electric and magnetic fields in the total field, \mathbf{D}_{tot} and \mathbf{H}_{tot} satisfies the following conditions:

$$\mathbf{D}_{tot} = \mathbf{D}_{inc} + \mathbf{D}_{scat} \quad (3.89)$$

$$\mathbf{H}_{tot} = \mathbf{H}_{inc} + \mathbf{H}_{scat} \quad (3.90)$$

where \mathbf{D}_{inc} and \mathbf{H}_{inc} are the electric and magnetic fields that are generated by the propagation of plane wave in the homogeneous free space in the absence of any object,

while the $\mathbf{D}_{_scat}$ and $\mathbf{H}_{_scat}$ are the electric and magnetic fields that are perturbation due to the presence of the object. The incident fields ($\mathbf{D}_{_inc}$ and $\mathbf{H}_{_inc}$), scattered fields ($\mathbf{D}_{_scat}$ and $\mathbf{H}_{_scat}$), and total fields ($\mathbf{D}_{_tot}$ and $\mathbf{H}_{_tot}$) separately satisfy the Maxwell's equations, respectively. And they can be solved using the ADI-FDTD scheme, respectively.

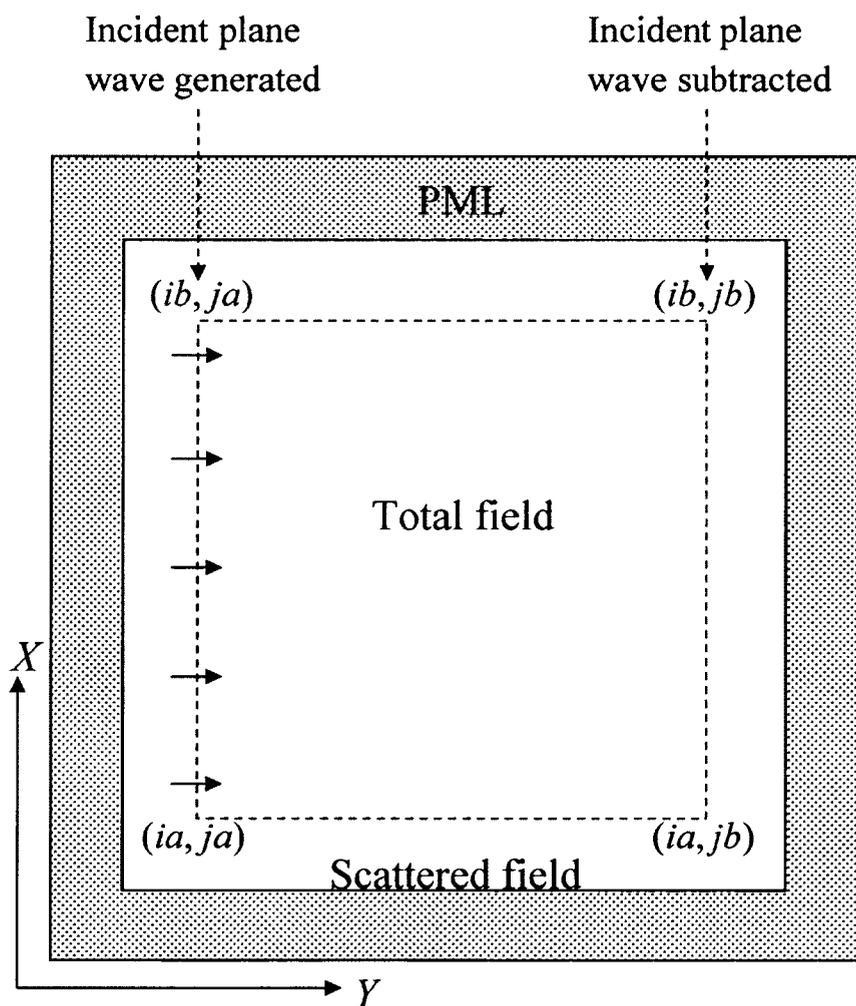


Figure 3.3 Total/Scattered field of the two-dimensional problem space

It is similar to set up the total/scattered field in the three-dimensional problem space, except that the boundaries between the total field and the scattered field are planes

instead of the lines. It should be pointed out the incident fields can simply be kept in one-dimensional arrays. Choosing a source point and adding the incident E_z at that point generates a plane wave. Here, an XZ plane wave is supposed to be generated at $j = ja$ and subtracted out at $j = jb$, as shown in Figure 3.4. Therefore, in the homogeneous free space without any object, there are only E_{z_inc} and H_{x_inc} .

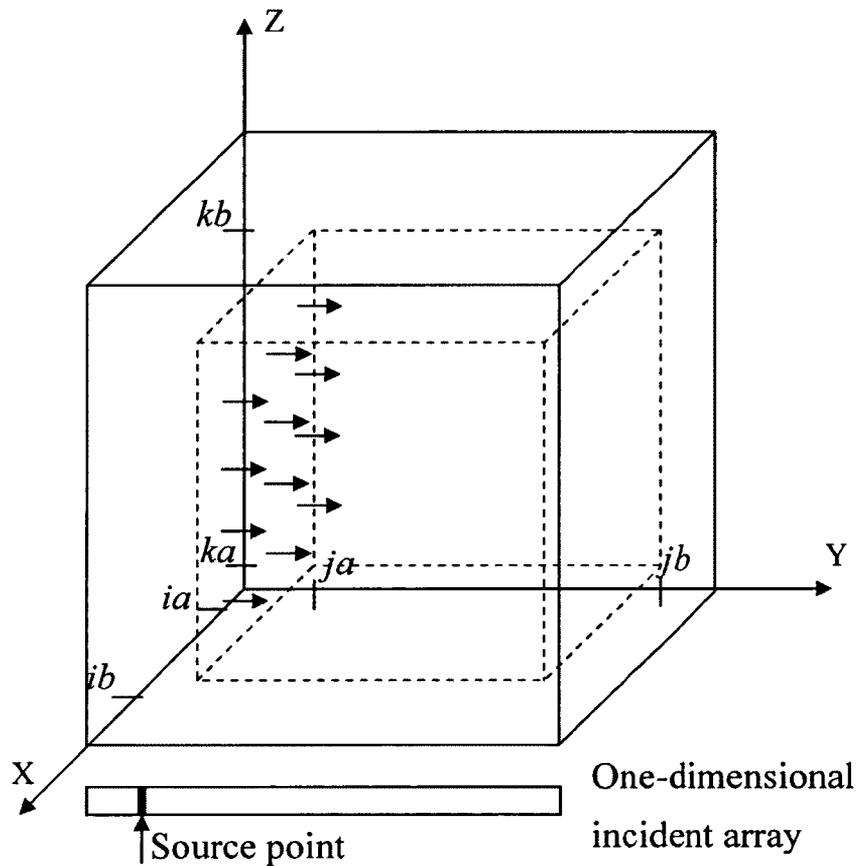


Figure 3.4 Total/Scattered field of the three-dimensional problem space

Since any point in the problem space is either in the total field or in the scattered field, if the updating of the field components of a point located in the total field uses the points in the scattered field, the updated value needs to be modified. Likewise, for the

points located in the scattered field, the same procedure is needed. Examining the ADI-FDTD scheme for a rectangular domain shows that there are eight places that must be modified. The detailed modifications are described as follows:

For the first half-time step:

1) D_y value at $k = ka$ or $k = kb$

$$D_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, ka) = D_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, ka) - \frac{c_0 \Delta t}{2\Delta z} H_{x_inc}^{n+\frac{1}{2}}(j + \frac{1}{2}). \quad (3.91)$$

$$D_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, kb) = D_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, kb) + \frac{c_0 \Delta t}{2\Delta z} H_{x_inc}^{n+\frac{1}{2}}(j + \frac{1}{2}). \quad (3.92)$$

2) D_z value at $j = ja$ or $j = jb$

$$D_z^{n+\frac{1}{2}}(i, ja, k + \frac{1}{2}) = D_z^{n+\frac{1}{2}}(i, ja, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2\Delta y} H_{x_inc}^n(ja - \frac{1}{2}). \quad (3.93)$$

$$D_z^{n+\frac{1}{2}}(i, jb, k + \frac{1}{2}) = D_z^{n+\frac{1}{2}}(i, jb, k + \frac{1}{2}) - \frac{c_0 \Delta t}{2\Delta y} H_{x_inc}^n(jb + \frac{1}{2}). \quad (3.94)$$

3) H_x value just outside $j = ja$ and $j = jb$

$$H_x^{n+\frac{1}{2}}(i, ja - \frac{1}{2}, k + \frac{1}{2}) = H_x^{n+\frac{1}{2}}(i, ja - \frac{1}{2}, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2\Delta y} E_{z_inc}^n(ja). \quad (3.95)$$

$$H_x^{n+\frac{1}{2}}(i, jb + \frac{1}{2}, k + \frac{1}{2}) = H_x^{n+\frac{1}{2}}(i, jb + \frac{1}{2}, k + \frac{1}{2}) - \frac{c_0 \Delta t}{2\Delta y} E_{z_inc}^n(jb). \quad (3.96)$$

4) H_y value just outside $i = ia$ and $i = ib$

$$H_y^{n+\frac{1}{2}}(ia - \frac{1}{2}, j, k + \frac{1}{2}) = H_y^{n+\frac{1}{2}}(ia - \frac{1}{2}, j, k + \frac{1}{2}) - \frac{c_0 \Delta t}{2\Delta x} E_{z_inc}^{n+\frac{1}{2}}(j). \quad (3.97)$$

$$H_y^{n+\frac{1}{2}}(ib + \frac{1}{2}, j, k + \frac{1}{2}) = H_y^{n+\frac{1}{2}}(ib + \frac{1}{2}, j, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2\Delta x} E_{z_inc}^{n+\frac{1}{2}}(j). \quad (3.98)$$

For the second half-time step:

1) D_y value at $k = ka$ or $k = kb$

$$D_y^{n+1}(i, j + \frac{1}{2}, ka) = D_y^{n+1}(i, j + \frac{1}{2}, ka) - \frac{c_0 \Delta t}{2\Delta z} H_{x_inc}^{n+\frac{1}{2}}(j + \frac{1}{2}). \quad (3.99)$$

$$D_y^{n+1}(i, j + \frac{1}{2}, kb) = D_y^{n+1}(i, j + \frac{1}{2}, kb) + \frac{c_0 \Delta t}{2\Delta z} H_{x_inc}^{n+\frac{1}{2}}(j + \frac{1}{2}). \quad (3.100)$$

2) D_z value at $j = ja$ or $j = jb$

$$D_z^{n+1}(i, ja, k + \frac{1}{2}) = D_z^{n+1}(i, ja, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2\Delta y} H_{x_inc}^{n+1}(ja - \frac{1}{2}). \quad (3.101)$$

$$D_z^{n+1}(i, jb, k + \frac{1}{2}) = D_z^{n+1}(i, jb, k + \frac{1}{2}) - \frac{c_0 \Delta t}{2\Delta y} H_{x_inc}^{n+1}(jb + \frac{1}{2}). \quad (3.102)$$

3) H_x value just outside $j = ja$ and $j = jb$

$$H_x^{n+1}(i, ja - \frac{1}{2}, k + \frac{1}{2}) = H_x^{n+1}(i, ja - \frac{1}{2}, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2\Delta y} E_{z_inc}^{n+1}(ja). \quad (3.103)$$

$$H_x^{n+1}(i, jb + \frac{1}{2}, k + \frac{1}{2}) = H_x^{n+1}(i, jb + \frac{1}{2}, k + \frac{1}{2}) - \frac{c_0 \Delta t}{2\Delta y} E_{z_inc}^{n+1}(jb). \quad (3.104)$$

4) H_y value just outside $i = ia$ and $i = ib$

$$H_y^{n+1}(ia - \frac{1}{2}, j, k + \frac{1}{2}) = H_y^{n+1}(ia - \frac{1}{2}, j, k + \frac{1}{2}) - \frac{c_0 \Delta t}{2\Delta x} E_{z_inc}^{n+\frac{1}{2}}(j). \quad (3.105)$$

$$H_y^{n+1}(ib + \frac{1}{2}, j, k + \frac{1}{2}) = H_y^{n+1}(ib + \frac{1}{2}, j, k + \frac{1}{2}) + \frac{c_0 \Delta t}{2\Delta x} E_{z_inc}^{n+\frac{1}{2}}(j). \quad (3.106)$$

3.5 Source Term

One of the most commonly used source terms in the FDTD simulation is the Gaussian pulse given by

$$f(t) = A_{mp} \cdot e^{-\frac{(t-t_0)^2}{2w^2}}, \quad (3.107)$$

where A_{mp} is the amplitude of the pulse, t_0 is the center of the pulse, and w controls the width of the pulse. However, the Gaussian pulse may be undesirable for some numerical simulation since it contains the direct current (DC) component [59], [61]. In order to eliminate the DC component, the differentiated Gaussian pulse can be used, which is obtained by taking a time derivative of the pure Gaussian pulse. The differentiated Gaussian pulse can be expressed as [61]

$$f(t) = -A_{mp} \frac{(t-t_0)}{w} \cdot e^{-\frac{(t-t_0)^2}{2w^2}}, \quad (3.108)$$

In this research, the differentiated Gaussian pulse as defined in Eq. (3.108) is employed to be the source term to numerically simulate the electromagnetic fields inside the biological matters when exposed to a nanopulse.

3.6 Computational Procedure

Based on the obtained numerical scheme in the previous sections, the computational procedure for simulating the electromagnetic fields inside the biological matters exposed to a nanopulse at time level $n+1$ from time level n can be described as follows:

Step 1. Set up g and f as defined in Eqs. (3.80)-(3.83).

Step 2. For the first half time step, calculate electric flux density $\mathbf{D}^{n+1/2}$ from Eq. (3.84) and corresponding expressions for the y and z directions using the Thomas algorithm, respectively.

Step 3. Modify $D_y^{n+1/2}$ and $D_z^{n+1/2}$ using Eqs. (3.91)-(3.94).

Step 4. Calculate electric field intensity $\mathbf{E}^{n+1/2}$ from Eq. (3.13) and update $\mathbf{I}^{n+1/2}$ and $\mathbf{S}_m^{n+1/2}$ ($m = 1, 2, 3, 4$) from Eqs. (3.16) and (3.17).

Step 5. Calculate magnetic field intensity $\mathbf{H}^{n+1/2}$ from Eq. (3.71) and corresponding expressions for the y and z directions.

Step 6. Modify $H_x^{n+1/2}$ and $H_y^{n+1/2}$ using Eqs. (3.95)-(3.98).

Step 7. For the second half time step, calculate electric flux density \mathbf{D}^{n+1} from the corresponding expressions similar to Eq. (3.84) for x , y , and z directions using the Thomas algorithm, respectively.

Step 8. Modify D_y^{n+1} and D_z^{n+1} using Eqs. (3.99)-(3.102).

Step 9. Calculate electric field intensity \mathbf{E}^{n+1} from Eq. (3.13) and update \mathbf{I}^{n+1} and \mathbf{S}_m^{n+1} ($m = 1, 2, 3, 4$) from Eqs. (3.16) and (3.17).

Step 10. Calculate magnetic field intensity \mathbf{H}^{n+1} from Eq. (3.73) and corresponding expressions for the y and z directions.

Step 11. Modify H_x^{n+1} and H_y^{n+1} using Eqs. (3.103)-(3.106).

Repeat steps 2-11 until the required time steps are reached.

CHAPTER FOUR

NUMERICAL EXAMPLES

In this chapter, two numerical examples are given to test the applicability of the proposed ADI-FDTD scheme. In the examples, both the ADI-FDTD scheme and the conventional FDTD scheme in [22] are employed in order to compare their results.

4.1 Example Description

The objective of this research is to develop a numerical scheme that can break through the CFL stability condition so that a larger time step can be used for the electromagnetic field simulations. In order to test the applicability of the proposed ADI-FDTD scheme, the electromagnetic fields induced by a differentiated Gaussian pulse and penetrating into two different biological matters are simulated. The pulse is chosen to be

$$E_z = -10^6 \frac{n-100}{\sqrt{200}} e^{-(n-100)^2/400} \text{ V/m}, \quad (4.1)$$

where n is the number of time step. The shape of the pulse is shown in Figure 4.1. A plane wave is generated by the differentiated Gaussian pulse in an XZ plane and propagated in the y direction. The biological tissue is located at the center of a cube with the dimension of $60 \mu\text{m} \times 60 \mu\text{m} \times 60 \mu\text{m}$. For both proposed ADI-FDTD scheme and the conventional FDTD scheme, a uniform mesh with $\Delta x = \Delta y = \Delta z = 1.0 \mu\text{m}$ is employed, which leads to a mesh of $60 \times 60 \times 60$ grid points. A five-point PML is

employed in all the simulations. $\Delta t_{FDTD} = \Delta x/2c_0 = 1.6667 \times 10^{-15}$ s is chosen in order to satisfy the CFL stability conditions. On the other hand, $2\Delta t_{FDTD}$ violates the CFL stability condition.

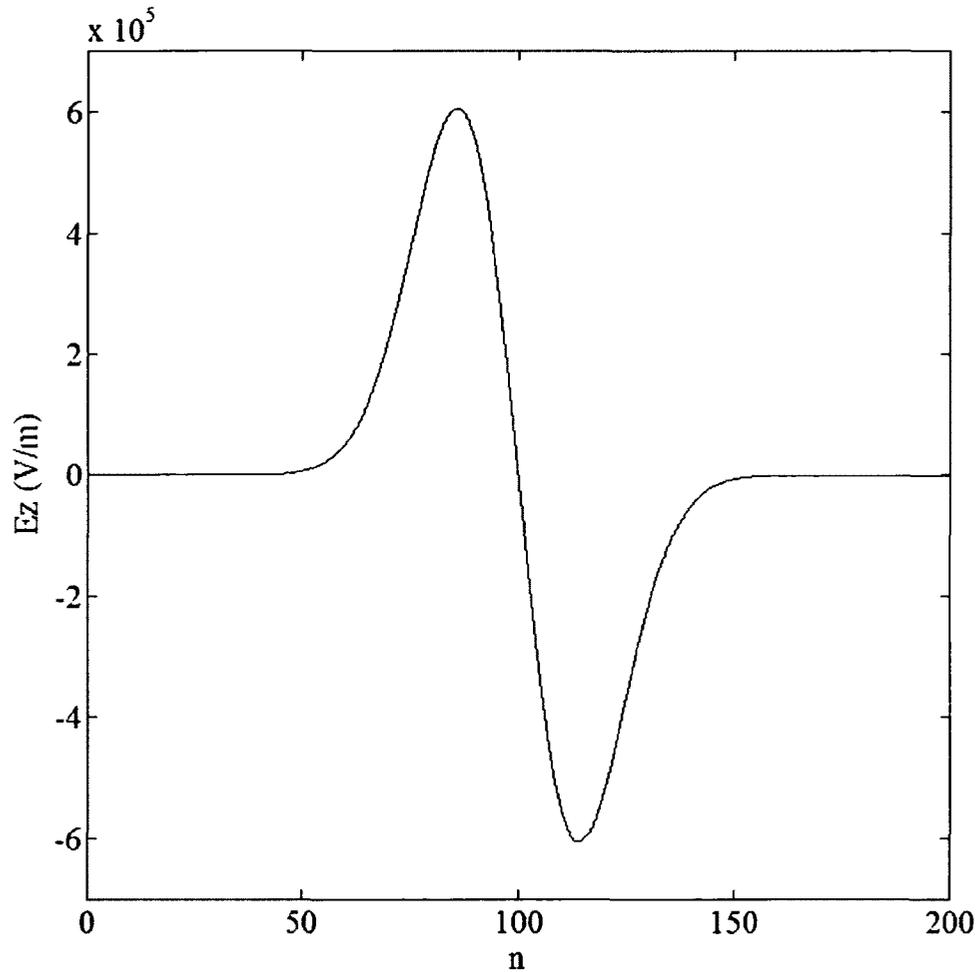


Figure 4.1 Pulse shape of $E_z = -10^6 \frac{n-100}{\sqrt{200}} e^{-(n-100)^2/400}$ V/m

There are two cases in the computation. Case 1 is to consider the biological tissue with the properties of breast fat. It is assumed to be a cube with side length of $20 \mu\text{m}$. In Case 2, the biological tissue is considered to be a blood cell. It is assumed to be a sphere

with the radius of 10 μm . The parameters in the Cole-Cole expression for these two cases are shown in Table 4.1.

Table 4.1 Dielectric properties [20], [48]

	Case 1	Case 2
	(Breast Fat)	(Blood)
ϵ_{∞}	2.5	4.0
σ_s	0.01	0.7
$\Delta\epsilon_1$	3	56
$\Delta\epsilon_2$	15	5200
$\Delta\epsilon_3$	5×10^4	0
$\Delta\epsilon_4$	5×10^7	0
τ_1 (ps)	17.680	8.377
τ_2 (ns)	63.660	132.629
τ_3 (μs)	454.700	159.155
τ_4 (ms)	13.260	15.915
α_1	0.1	0.1
α_2	0.1	0.1
α_3	0.1	0.2
α_4	0	0

For both cases, the simulations are run for the conventional FDTD method using only two different time steps, Δt_{FDTD} and $2\Delta t_{FDTD}$, since it produces a divergent solution when $2\Delta t_{FDTD}$ is used. On the other hand, the simulations are run for the ADI-FDTD scheme using five different time steps, Δt_{FDTD} , $2\Delta t_{FDTD}$, $4\Delta t_{FDTD}$, $6\Delta t_{FDTD}$, and $8\Delta t_{FDTD}$, to see how large a time step can be chosen.

4.2 Results and Discussions

4.2.1 Numerical Verification of the Stability

Figures 4.2 and 4.3 show the simulation results of Case 1 obtained by the conventional FDTD scheme. Figure 4.2 is plotted based on the results with the time step to be Δt_{FDTD} . It shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 120, 144, 168, 192, and 216 time steps, respectively. This figure illustrates how the pulse touches the biological tissue, passes through the biological tissue, and passes over the biological tissue. It can be clearly seen that the frequency-dependent properties of the biological tissue, breast fat, affect the propagation of the nanopulse in the problem space. Figure 4.3 shows the results obtained with the time step to be $2\Delta t_{FDTD}$. It shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 5, 10, and 15 time steps, respectively. As can be seen, the solution with $2\Delta t_{FDTD}$ becomes more and more large and eventually divergent. This figure indicates that the conventional FDTD scheme will become numerically unstable if the time step does not satisfy the CFL stability condition.

Figures 4.4-4.8 show the simulation results of Case 1 obtained by the proposed ADI-FDTD scheme with time step to be Δt_{FDTD} , $2\Delta t_{FDTD}$, $4\Delta t_{FDTD}$, $6\Delta t_{FDTD}$, and $8\Delta t_{FDTD}$,

respectively. Figure 4.4 shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 120, 144, 168, 192, and 216 time steps, respectively. When the time step is doubled, the numbers of iterations needed for a certain time period is reduced by half. Figure 4.5 shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 60, 72, 84, 96, and 108 time steps, respectively. Figure 4.6 shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 30, 36, 42, 48, and 54 time steps, respectively. Figure 4.7 shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 20, 24, 28, 32, and 36 time steps, respectively. Figure 4.8 shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 15, 18, 21, 24, and 27 time steps, respectively. As can be seen, the results obtained by ADI-FDTD scheme remain stable with Δt_{FDTD} , $2\Delta t_{FDTD}$, $4\Delta t_{FDTD}$, and $6\Delta t_{FDTD}$, while the results with $8\Delta t_{FDTD}$ show small oscillations. This is probably caused by the PML since an empirical formula for PML is employed in this research and needs to be further investigated. Figures 4.4-4.8 imply that the larger time steps, which violate the CFL stability condition, can be employed in the proposed ADI-FDTD scheme.

Figures 4.9 and 4.10 show the simulation results of Case 2 obtained by the conventional FDTD scheme. Figure 4.9 is plotted based on the results with the time step to be Δt_{FDTD} . It shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 120, 144, 168, 192, and 216 time steps, respectively. This figure illustrates how the frequency-dependent properties of the blood cell affect the propagation of the nanopulse in the problem space. Figure 4.10 shows the results obtained with the time step to be $2\Delta t_{FDTD}$. It shows the contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after 5, 10, and 15 time steps. The drawback of the conventional FDTD scheme is seen again.

Figures 4.11-4.15 show the simulation results of Case 2 obtained by the proposed ADI-FDTD scheme with time step to be Δt_{FDTD} , $2\Delta t_{FDTD}$, $4\Delta t_{FDTD}$, $6\Delta t_{FDTD}$, and $8\Delta t_{FDTD}$, respectively. The numbers of time step are selected according to the time step used for each situation. Similar to Case 1, stable solutions can be obtained with Δt_{FDTD} , $2\Delta t_{FDTD}$, $4\Delta t_{FDTD}$, and $6\Delta t_{FDTD}$, while the solution with $8\Delta t_{FDTD}$ show small oscillations.

Figures 4.2-4.15 imply the frequency-dependent properties of the nanopulse penetration of biological Matters. Since breast fat and the blood cell used in these numerical simulations have different parameters for the Cole-Cole expression, this indicates that they influence the propagation of nanopulse in different way.

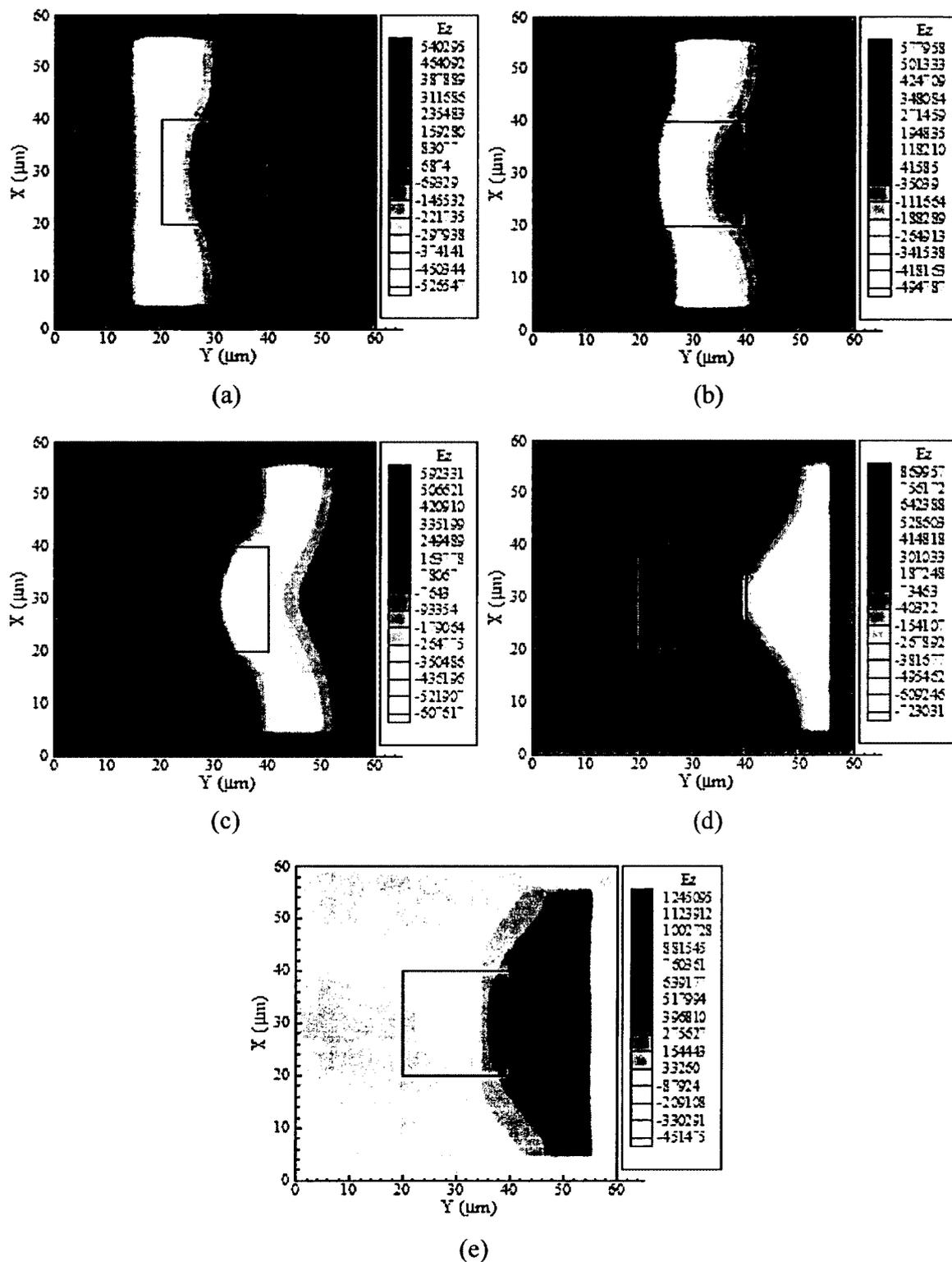


Figure 4.2 Simulation results of Case 1 obtained by the conventional FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.

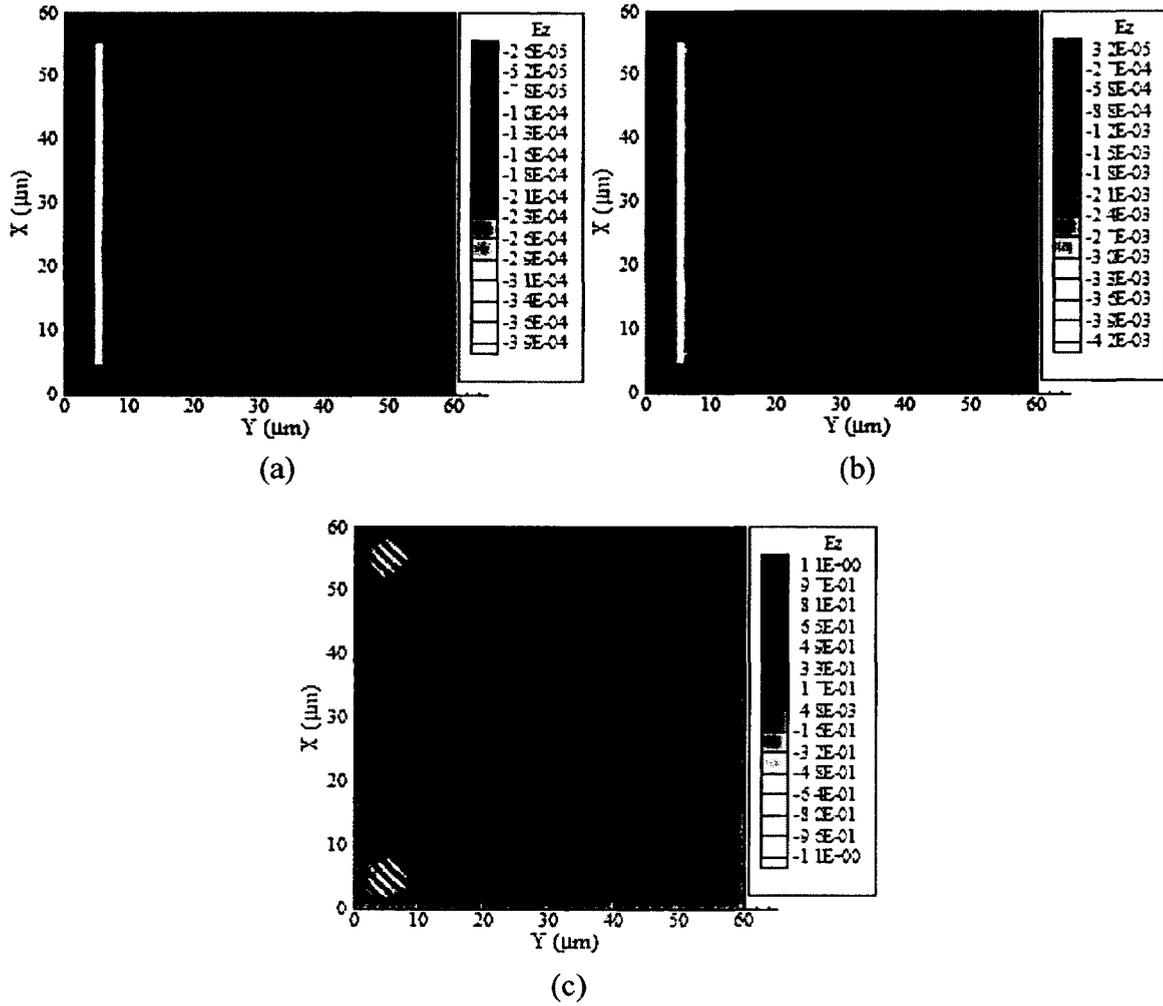


Figure 4.3 Simulation results of Case 1 obtained by the conventional FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 5 time steps, (b) 10 time steps, and (c) 15 time steps.

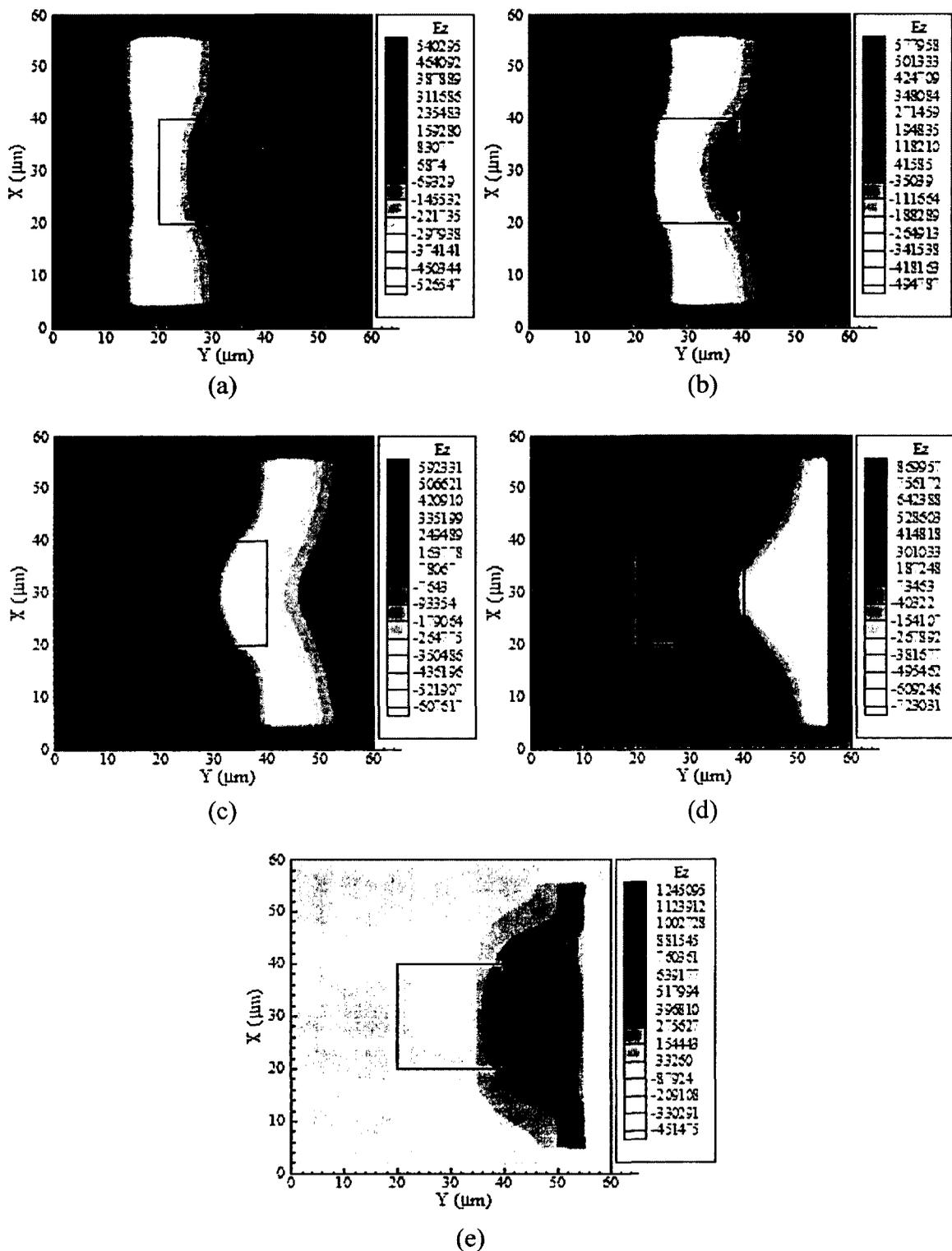


Figure 4.4 Simulation results of Case 1 obtained by the ADI-FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.

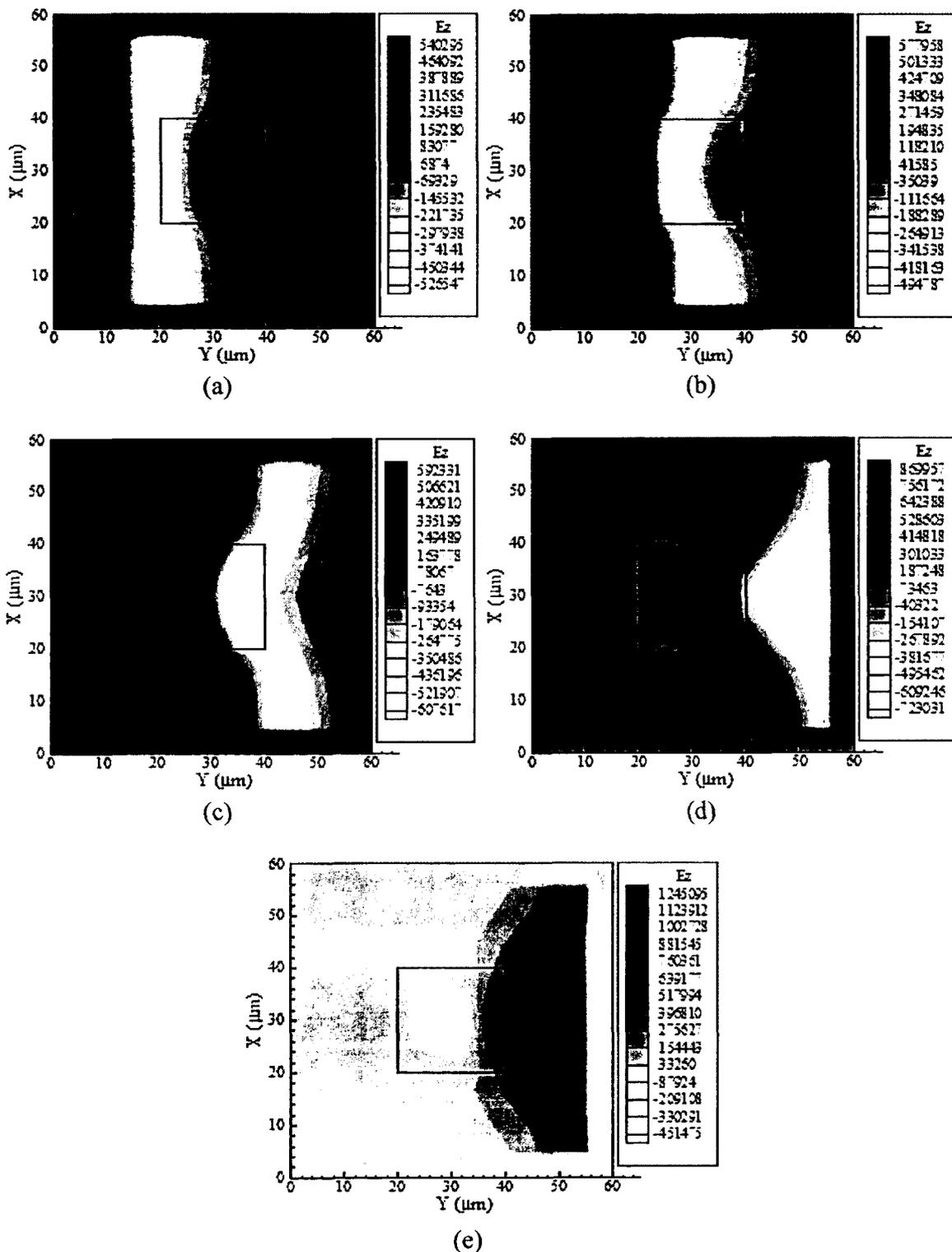


Figure 4.5 Simulation results of Case 1 obtained by the ADI-FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 60 time steps, (b) 72 time steps, (c) 83 time steps, (d) 96 time steps, and (e) 108 time steps.

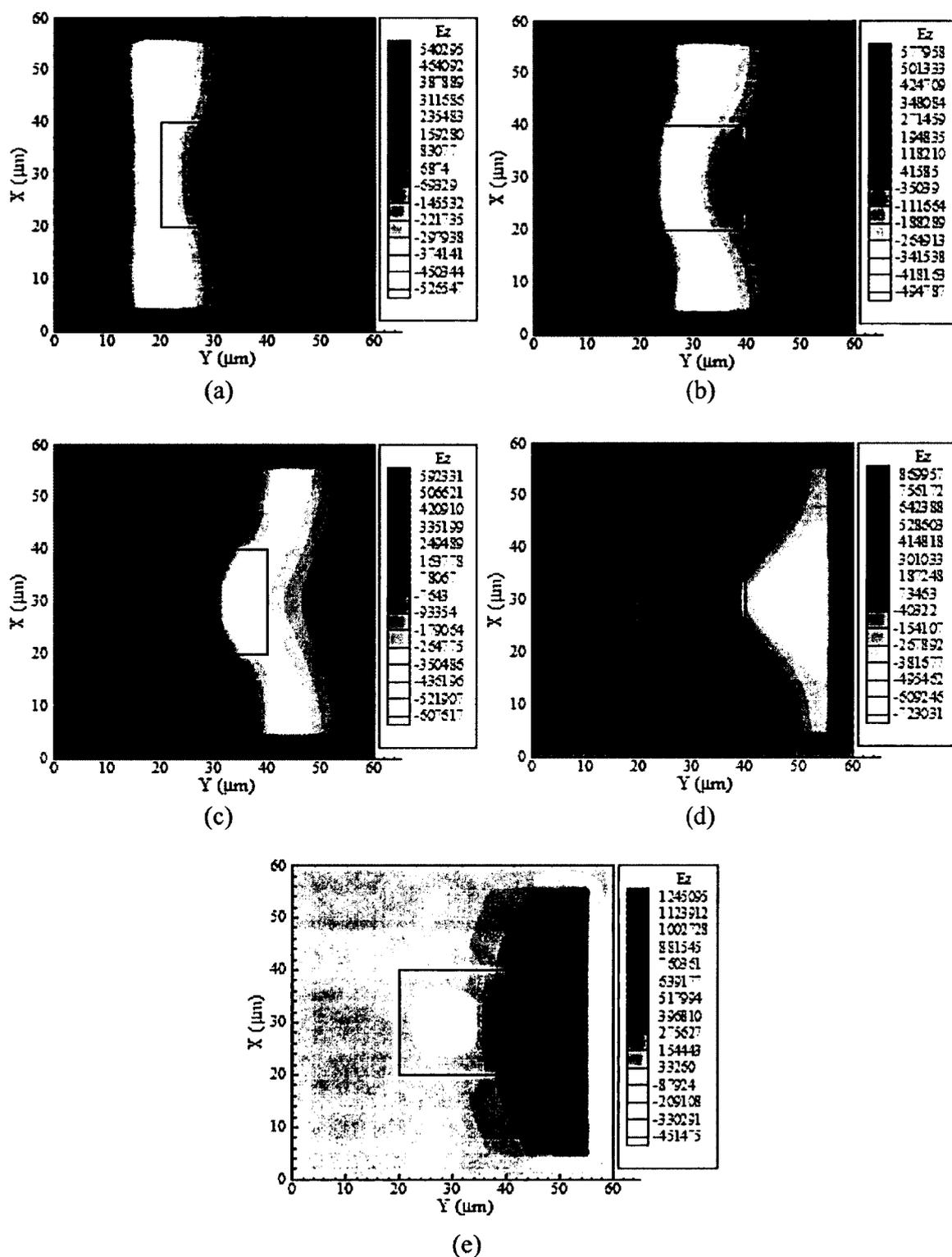


Figure 4.6 Simulation results of Case 1 obtained by the ADI-FDTD scheme with $4\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 30 time steps, (b) 36 time steps, (c) 42 time steps, (d) 48 time steps, and (e) 54 time steps.

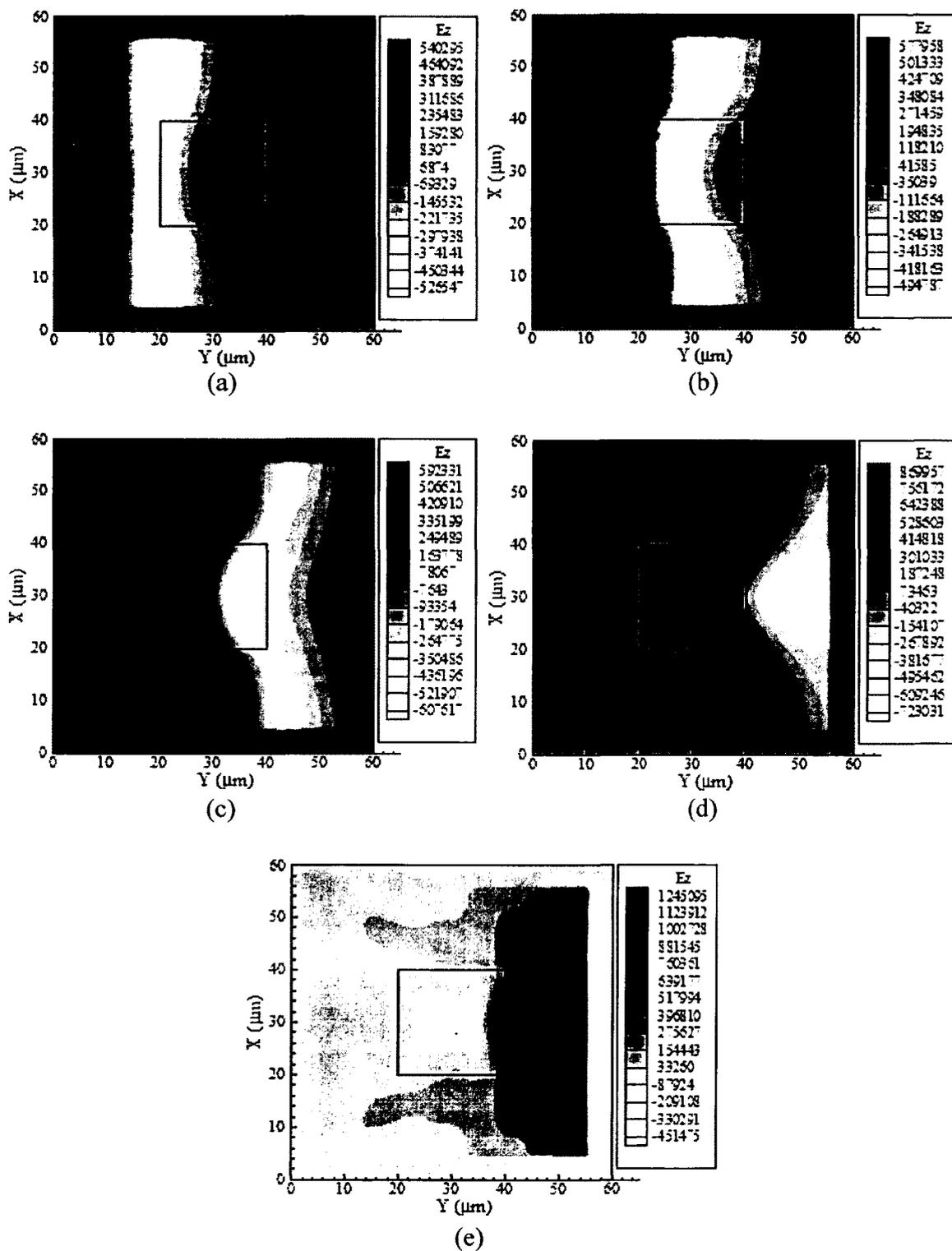


Figure 4.7 Simulation results of Case 1 obtained by the ADI-FDTD scheme with $6\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 20 time steps, (b) 24 time steps, (c) 28 time steps, (d) 32 time steps, and (e) 36 time steps.

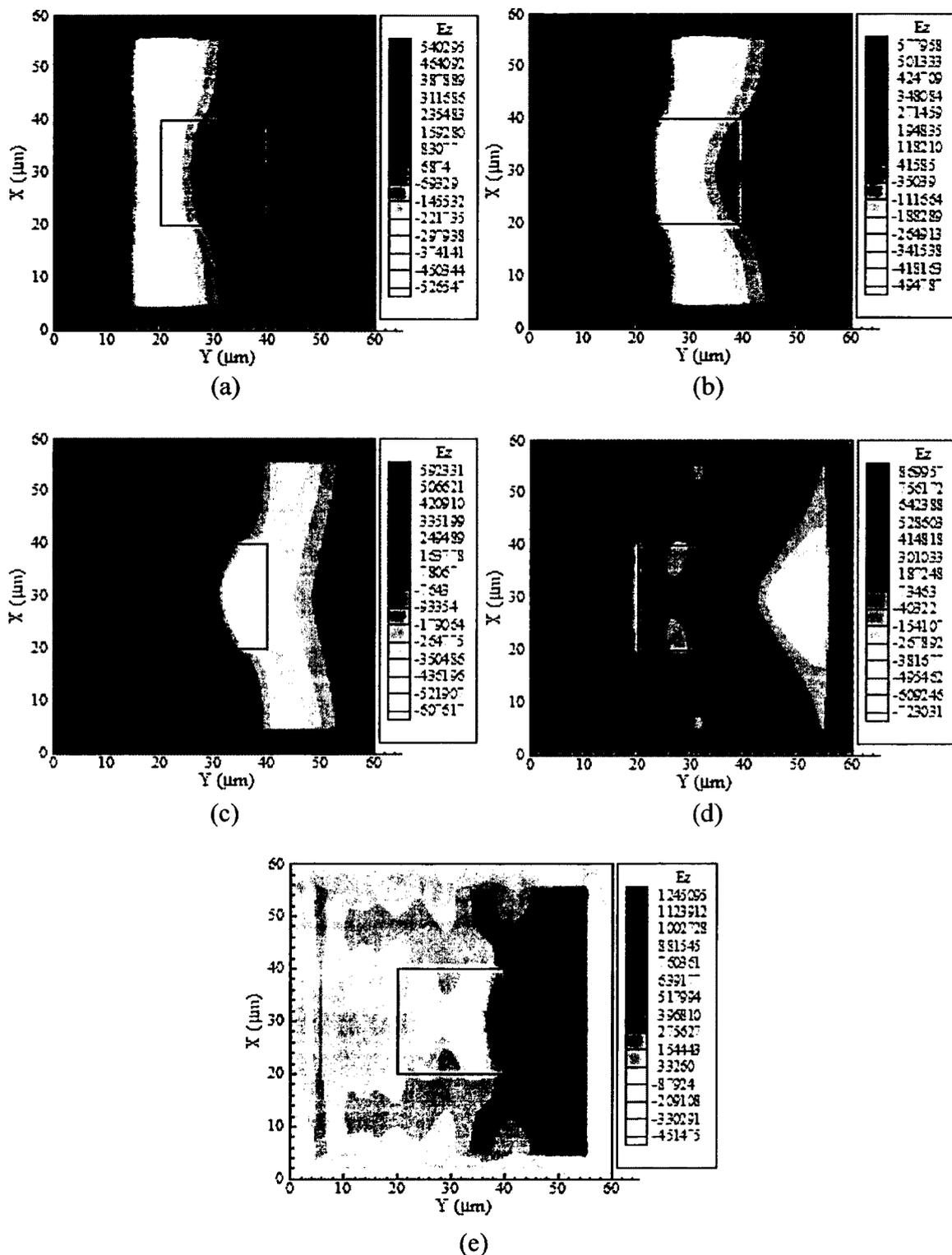


Figure 4.8 Simulation results of Case 1 obtained by the ADI-FDTD scheme with $8\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 15 time steps, (b) 18 time steps, (c) 21 time steps, (d) 24 time steps, and (e) 27 time steps.

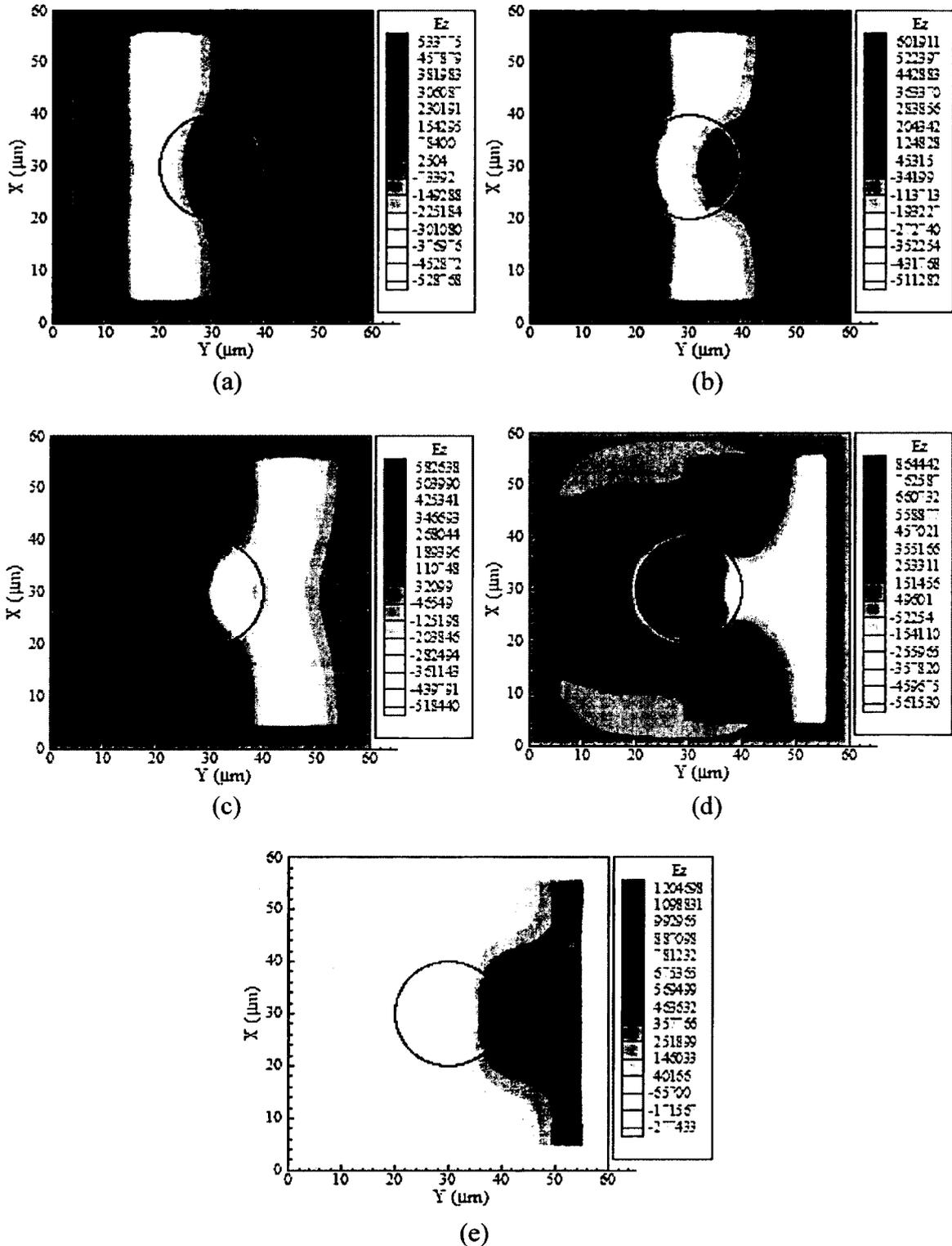


Figure 4.9 Simulation results of Case 2 obtained by the conventional FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.

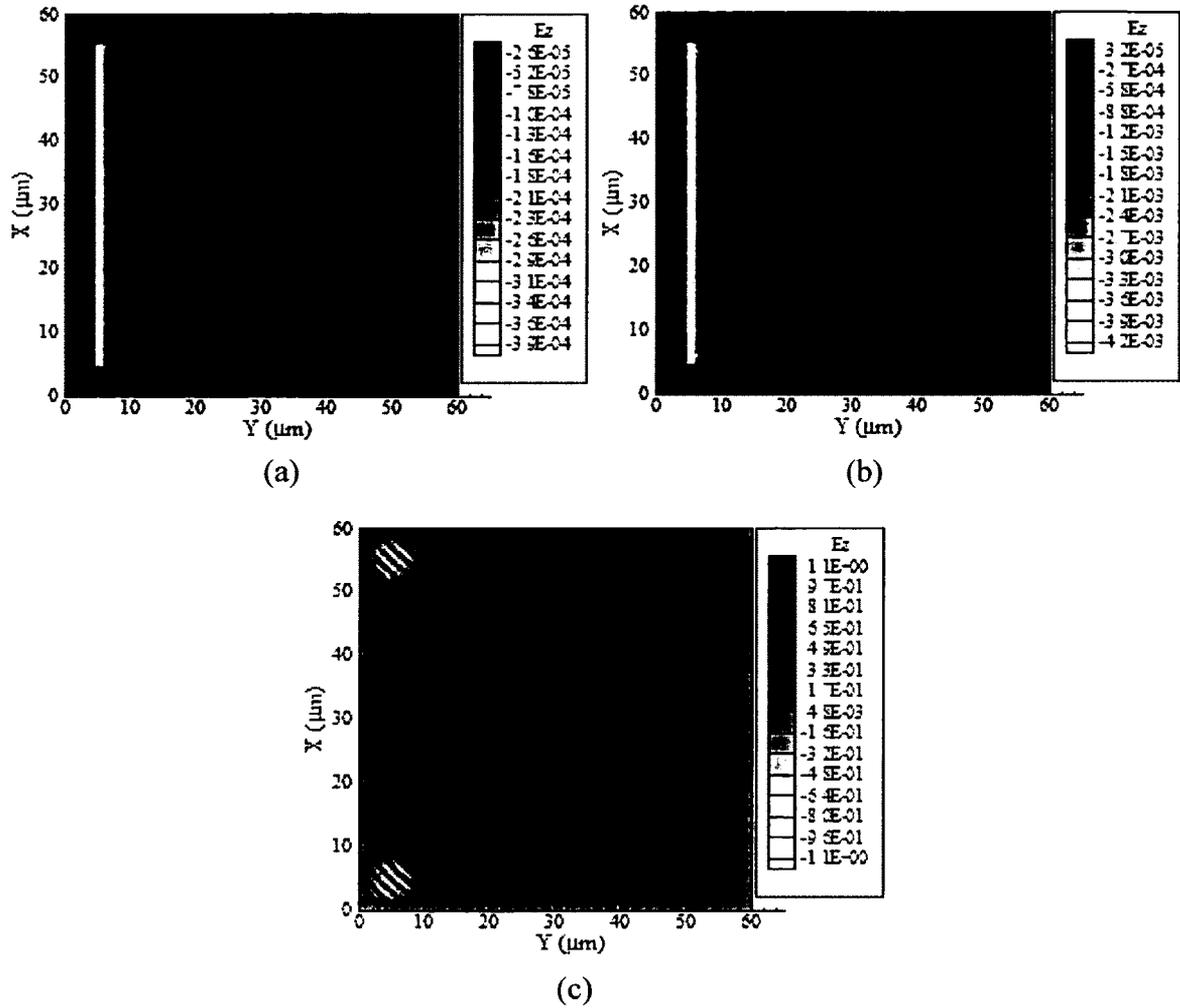


Figure 4.10 Simulation results of Case 2 obtained by the conventional FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 5 time steps, (b) 10 time steps, and (c) 15 time steps.

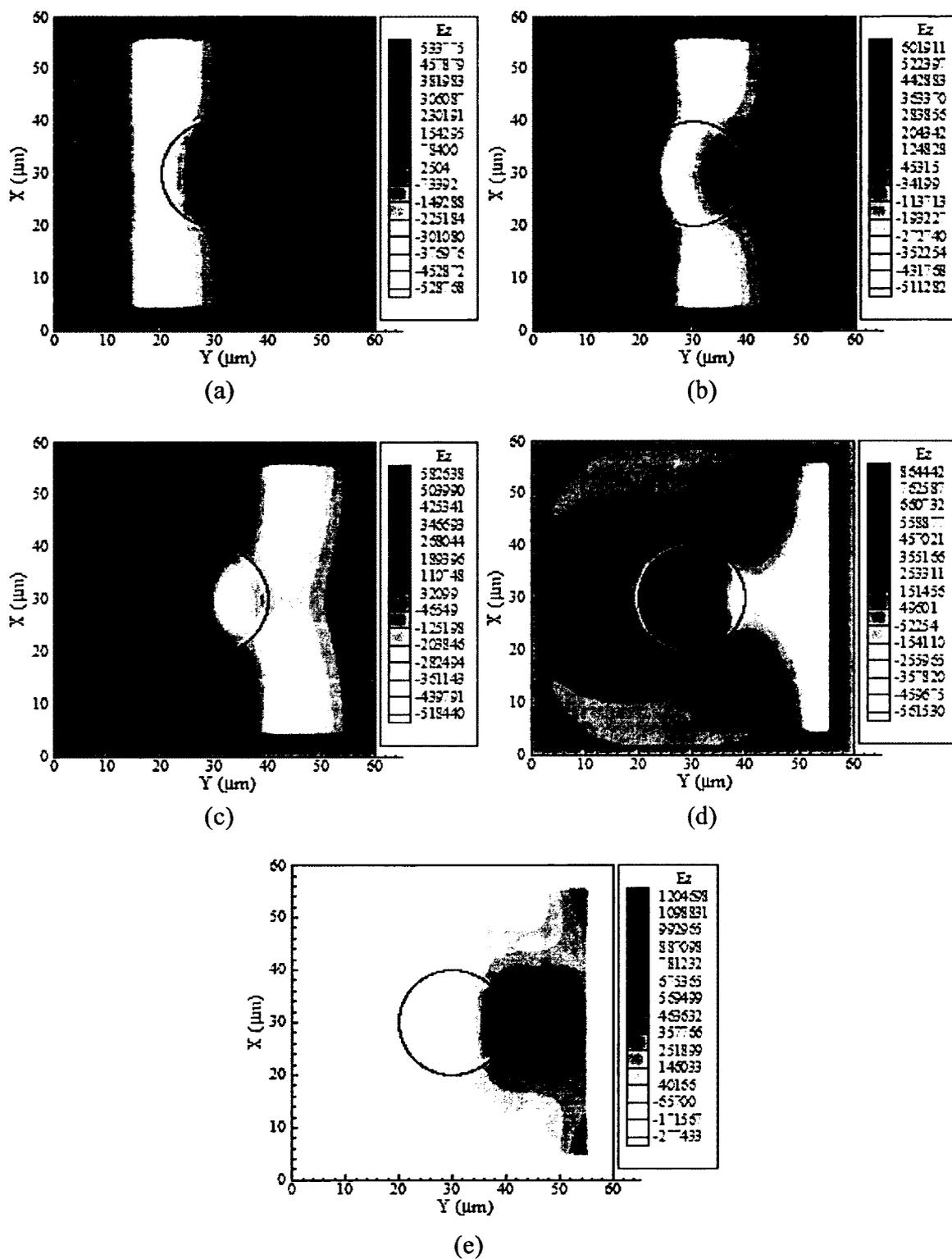


Figure 4.11 Simulation results of Case 2 obtained by the ADI-FDTD scheme with Δt_{FDTD} . Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 120 time steps, (b) 144 time steps, (c) 168 time steps, (d) 192 time steps, and (e) 216 time steps.

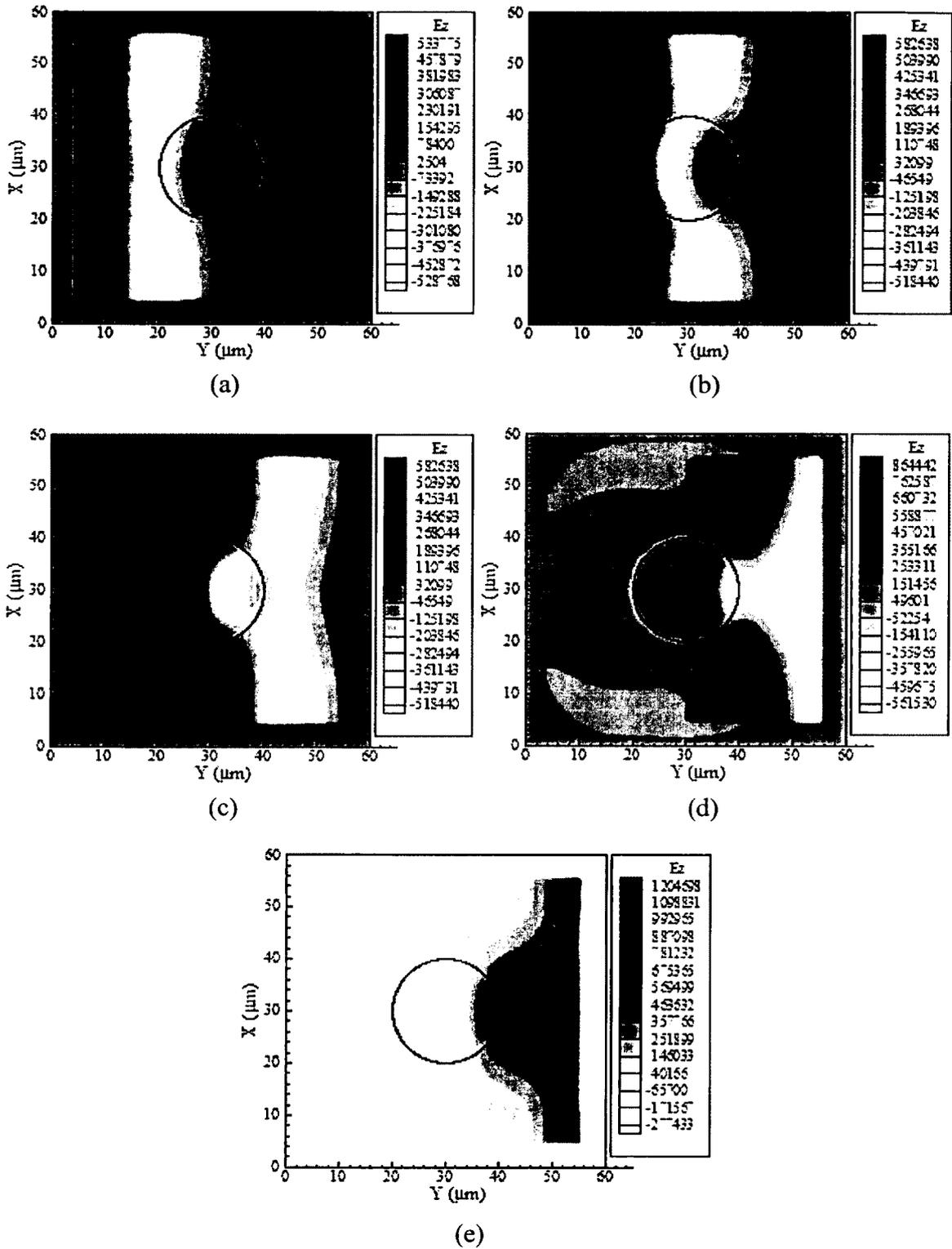


Figure 4.12 Simulation results of Case 2 obtained by the ADI-FDTD scheme with $2\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 60 time steps, (b) 72 time steps, (c) 84 time steps, (d) 96 time steps, and (e) 108 time steps.

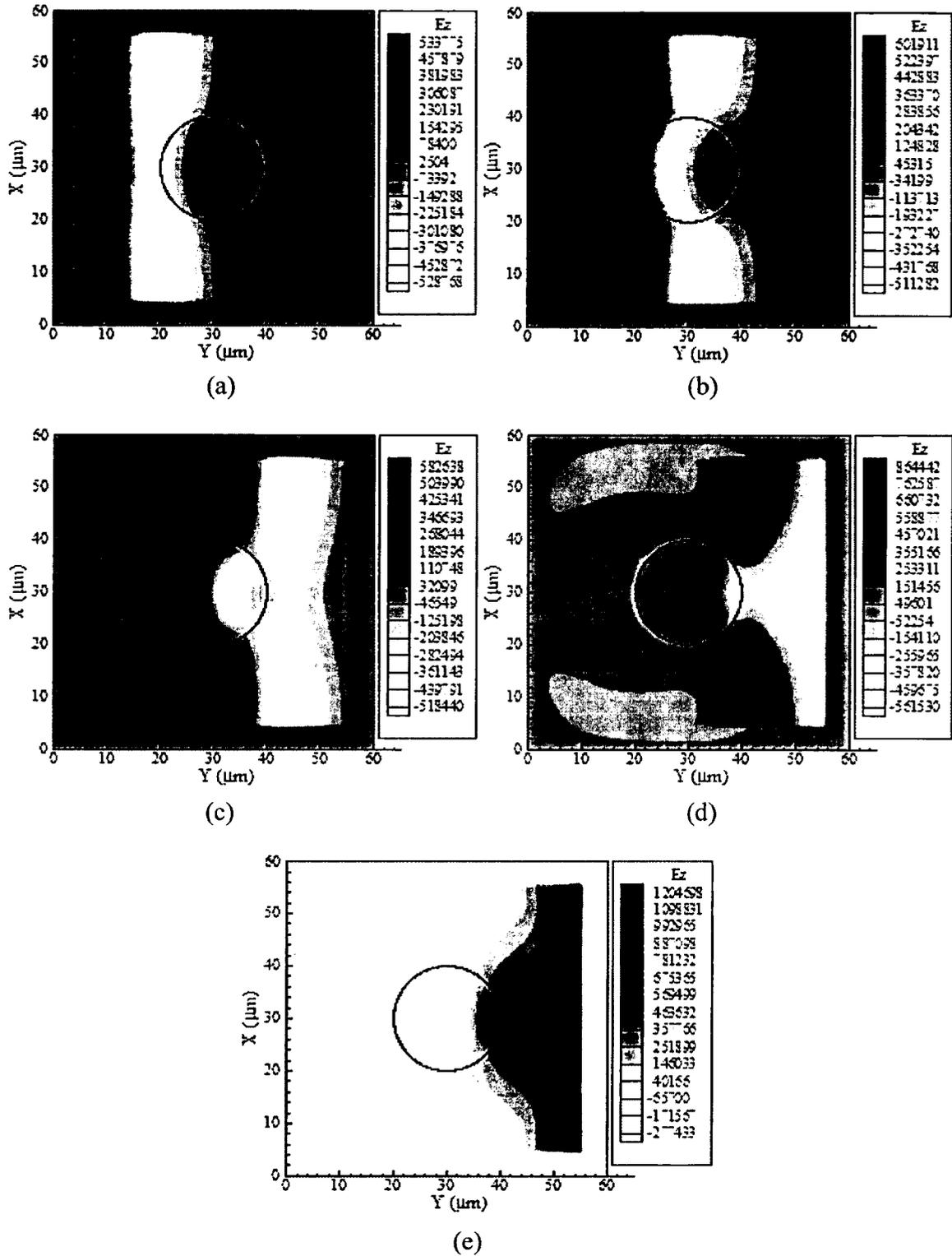


Figure 4.13 Simulation results of Case 2 obtained by the ADI-FDTD scheme with $4\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 30 time steps, (b) 36 time steps, (c) 42 time steps, (d) 48 time steps, and (e) 54 time steps.

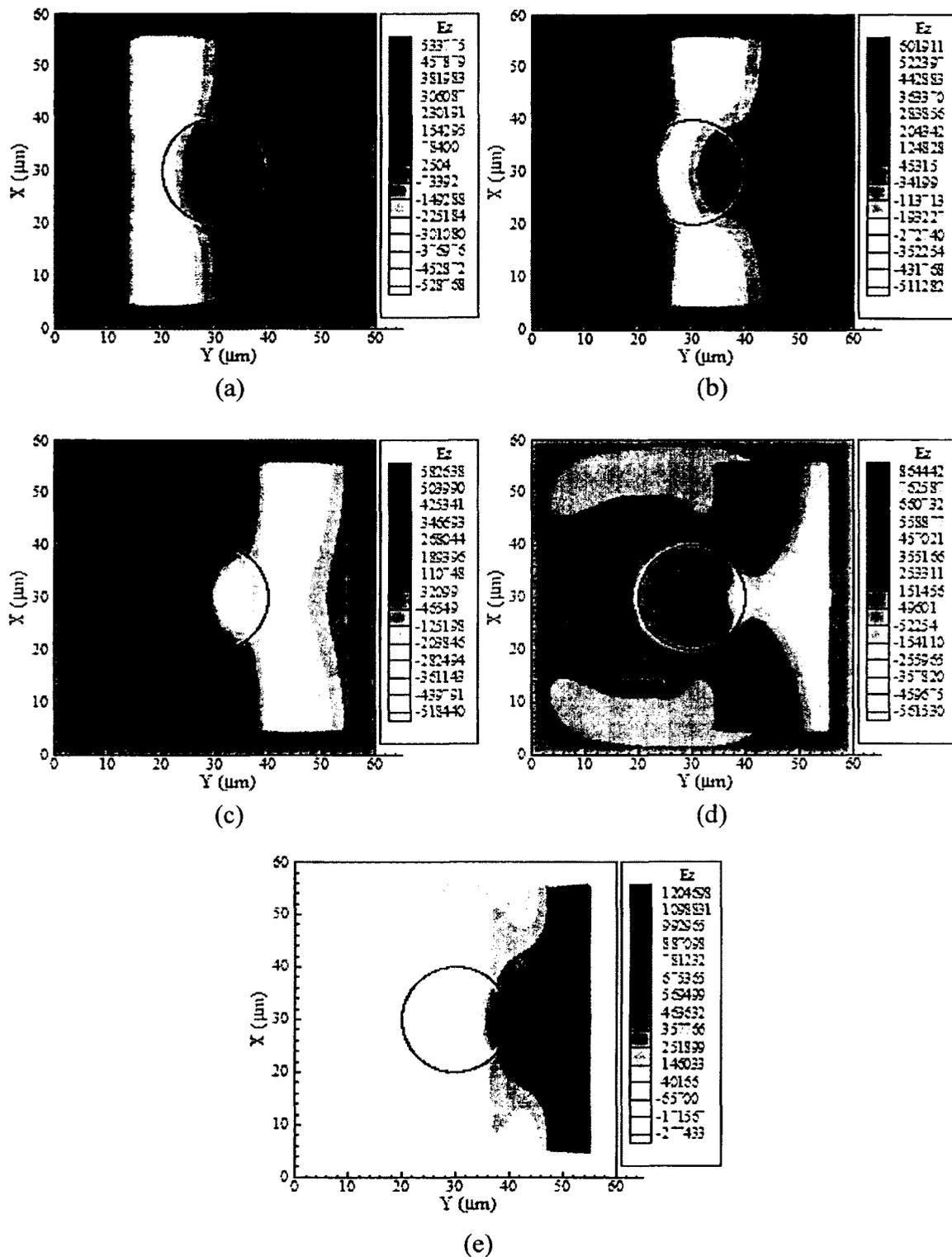


Figure 4.14 Simulation results of Case 2 obtained by the ADI-FDTD scheme with $6\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 20 time steps, (b) 24 time steps, (c) 28 time steps, (d) 32 time steps, and (e) 36 time steps.

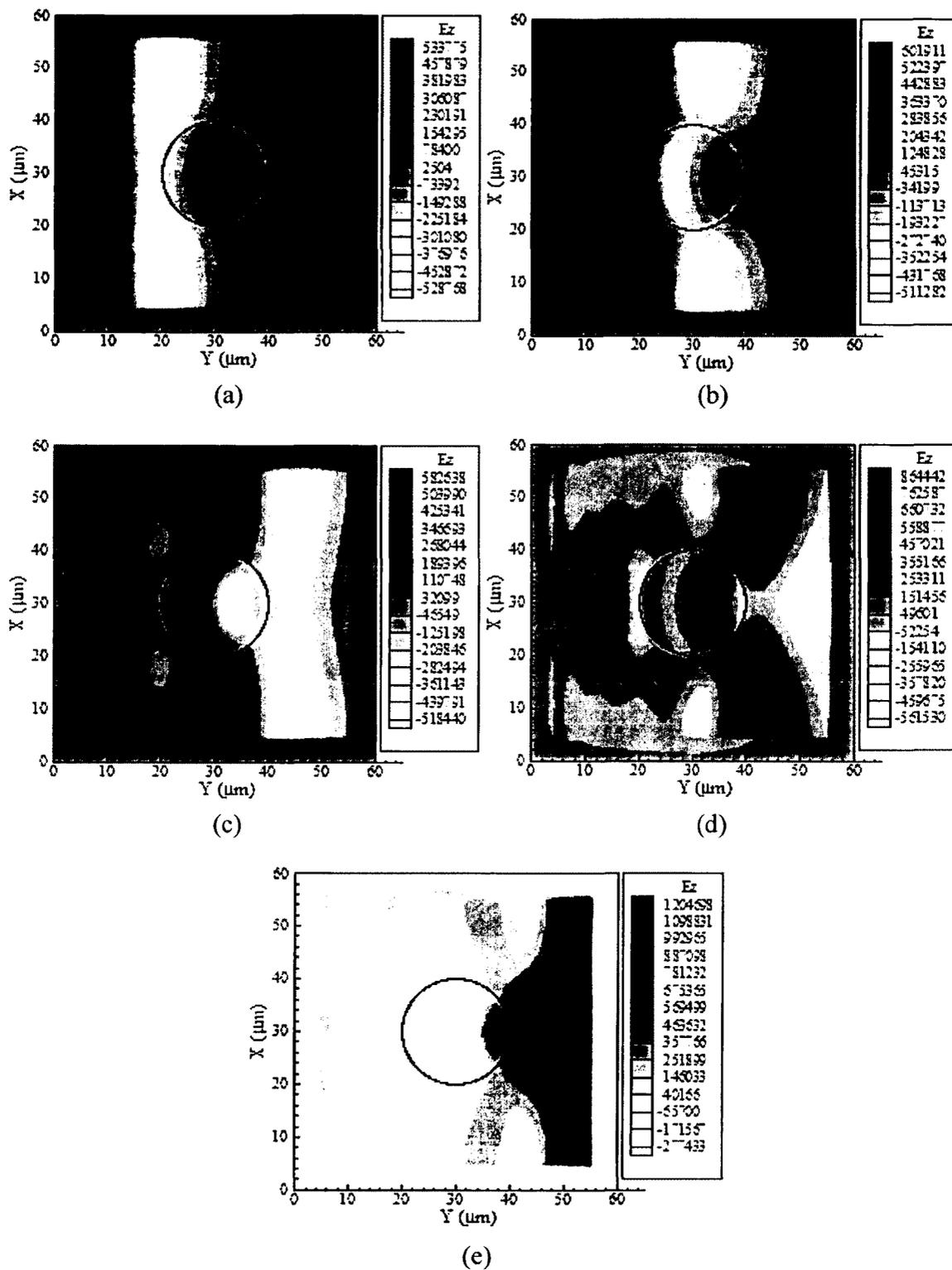


Figure 4.15 Simulation results of Case 2 obtained by the ADI-FDTD scheme with $8\Delta t_{FDTD}$. Contours of E_z in the xy cross-section at $z = 30 \mu\text{m}$ after (a) 15 time steps, (b) 18 time steps, (c) 21 time steps, (d) 24 time steps, and (e) 27 time steps.

4.2.2 Numerical Accuracy Versus Time Step

In the previous section, the stability of the proposed ADI-FDTD scheme has been tested by two numerical examples, indicating that the larger time steps can be used in the numerical simulations of electromagnetic fields. In this section, the numerical accuracy of the proposed ADI-FDTD scheme will be investigated since the modeling accuracy is very important for a numerical scheme. Because the conventional FDTD scheme has been widely used for solving the problems related to the electromagnetic fields, the results obtained by the conventional FDTD are used as the standard for the following comparisons.

For the purpose of comparison, the E_z values along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ were chosen in Case 1. Figure 4.16 shows a sequence of snapshots of the E_z values versus y position obtained by the conventional FDTD scheme with the time step to be Δt_{FDTD} and the ADI-FDTD scheme with five different time steps after various time. It can be seen that for a small time step, there are not significant differences between the conventional FDTD method and the ADI-FDTD method. With the time step increases, the difference becomes visible. This is because the truncation error with respect to time increases. When the time step is $8\Delta t_{FDTD}$, the difference becomes significant because of oscillation solutions.

For Case 2, the E_z values along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ were also chosen. Figure 4.17 shows a sequence of snapshots of the E_z values versus y position after different numbers of time steps. Similar to Figure 4.16, Figure 4.17 shows the comparison between the results obtained by the conventional FDTD scheme with the

time step to be Δt_{FDTD} and the ADI-FDTD scheme with time steps to be Δt_{FDTD} , $2\Delta t_{FDTD}$, $4\Delta t_{FDTD}$, $6\Delta t_{FDTD}$, and $8\Delta t_{FDTD}$. Results similar to those in Figure 4.16 can be seen, except the intensity of penetration is different due to the different properties of breast fat and blood.

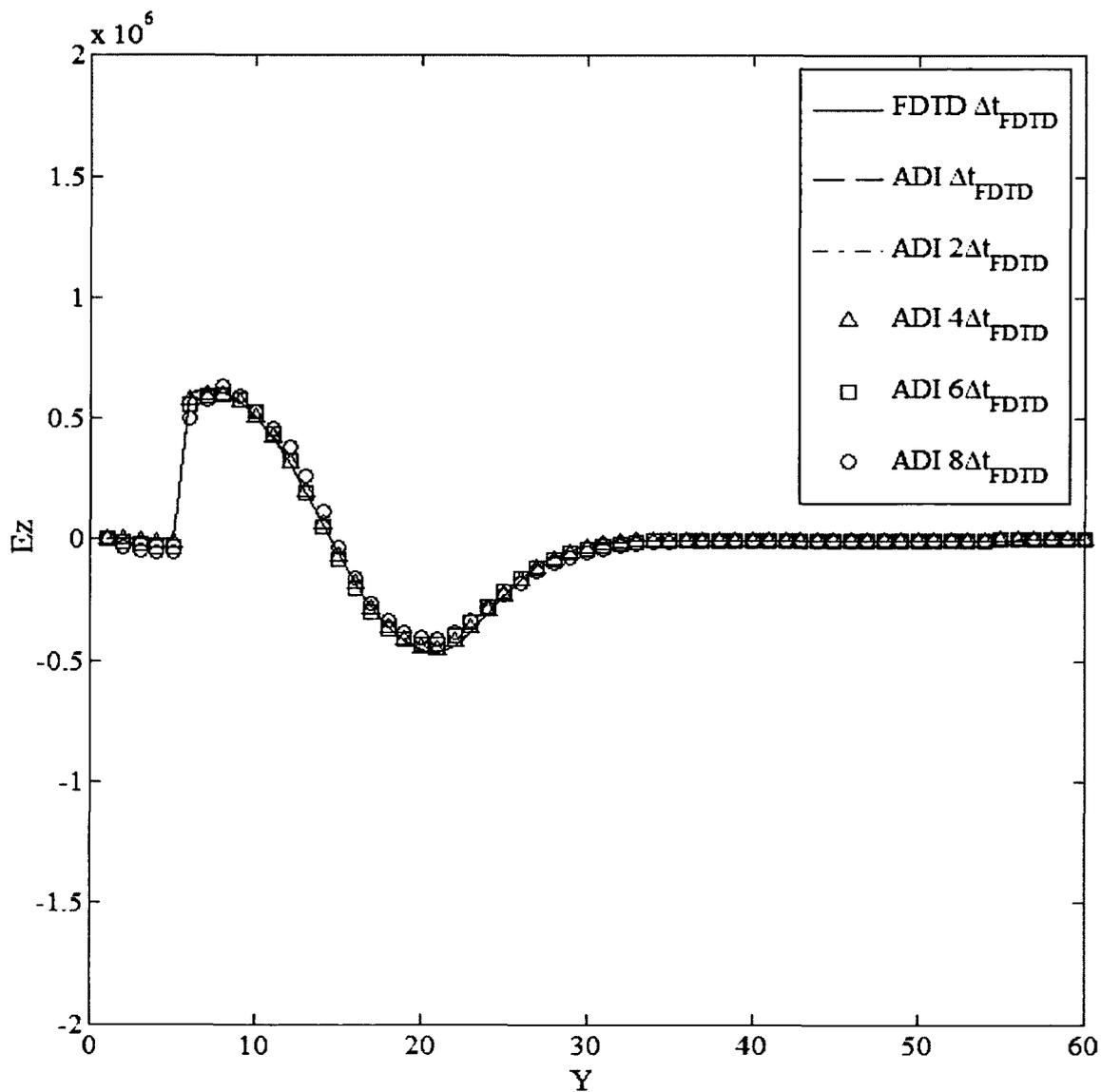


Figure 4.16a Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $120\Delta t_{FDTD}$.

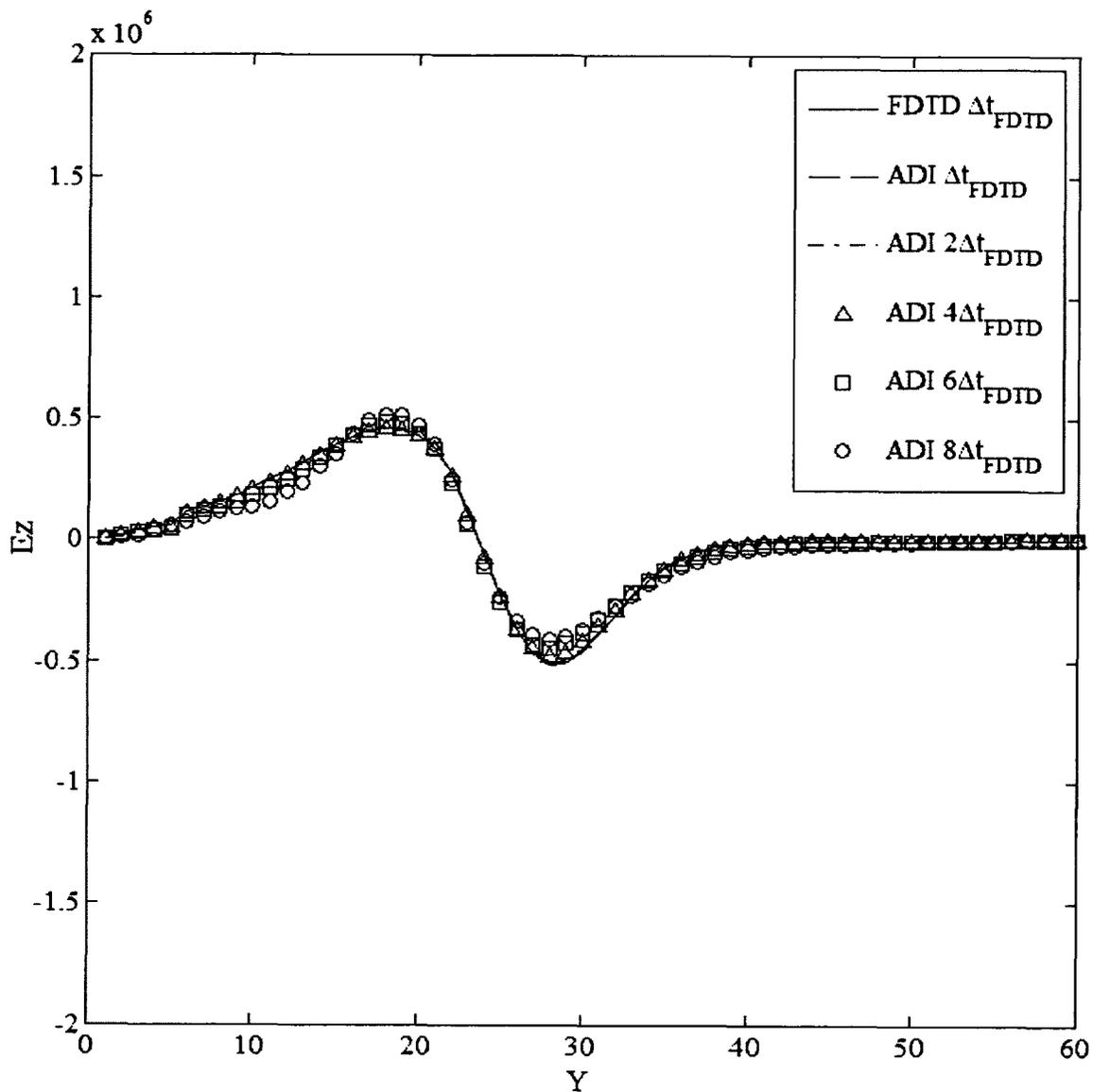


Figure 4.16b Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $144\Delta t_{FDTD}$.

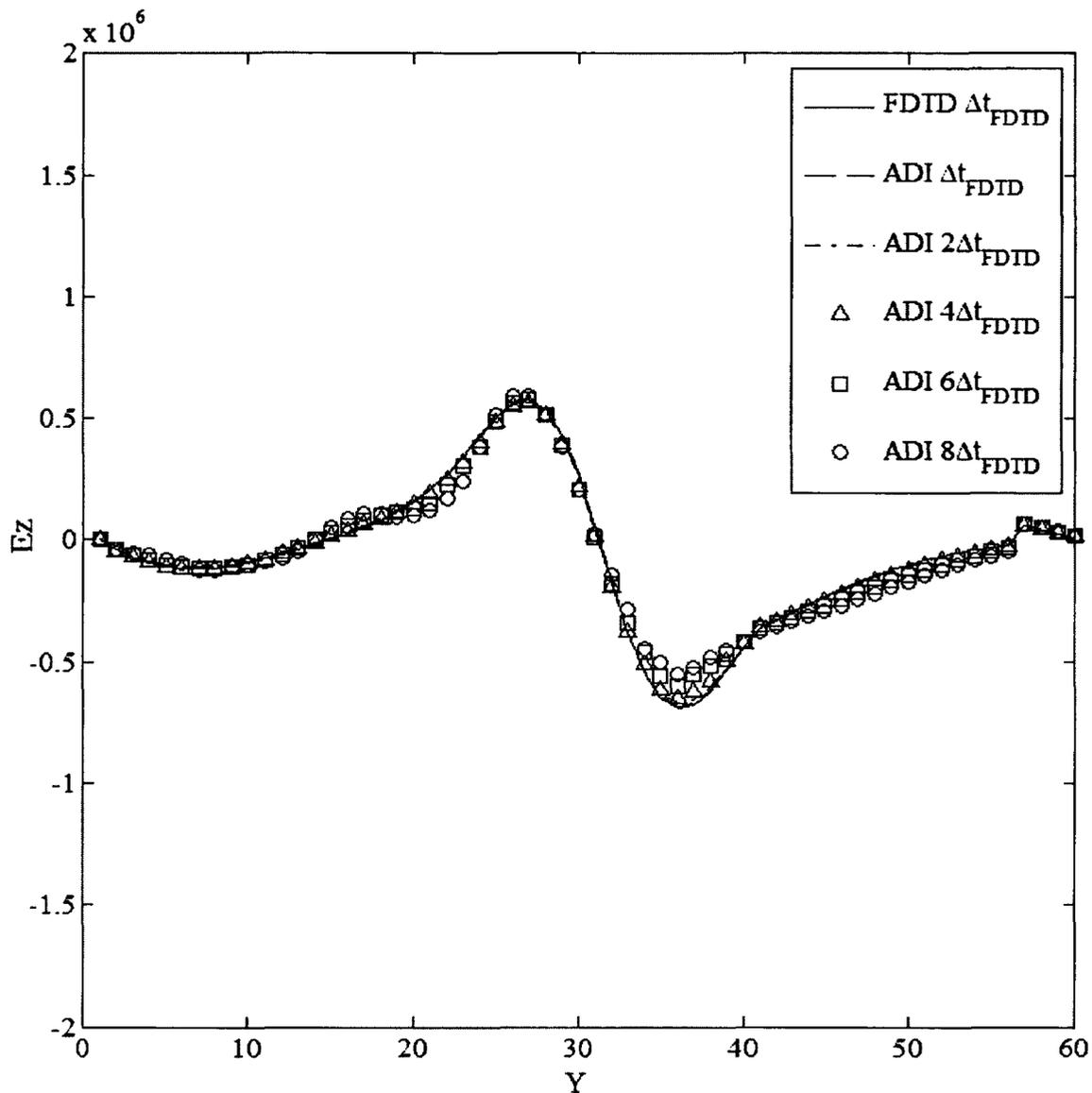


Figure 4.16c Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $168\Delta t_{\text{FDTD}}$.

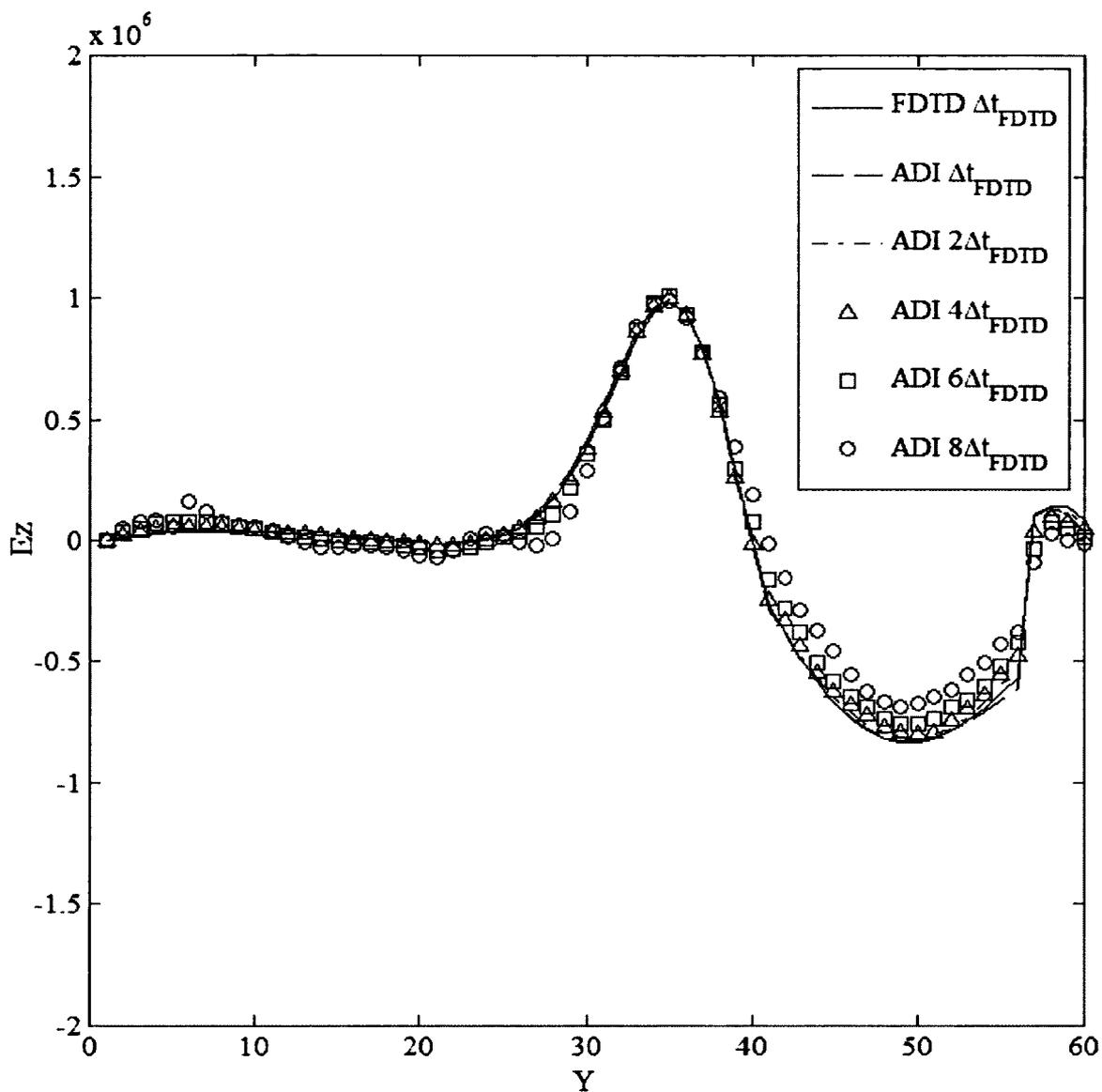


Figure 4.16d Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $192\Delta t_{FDTD}$.

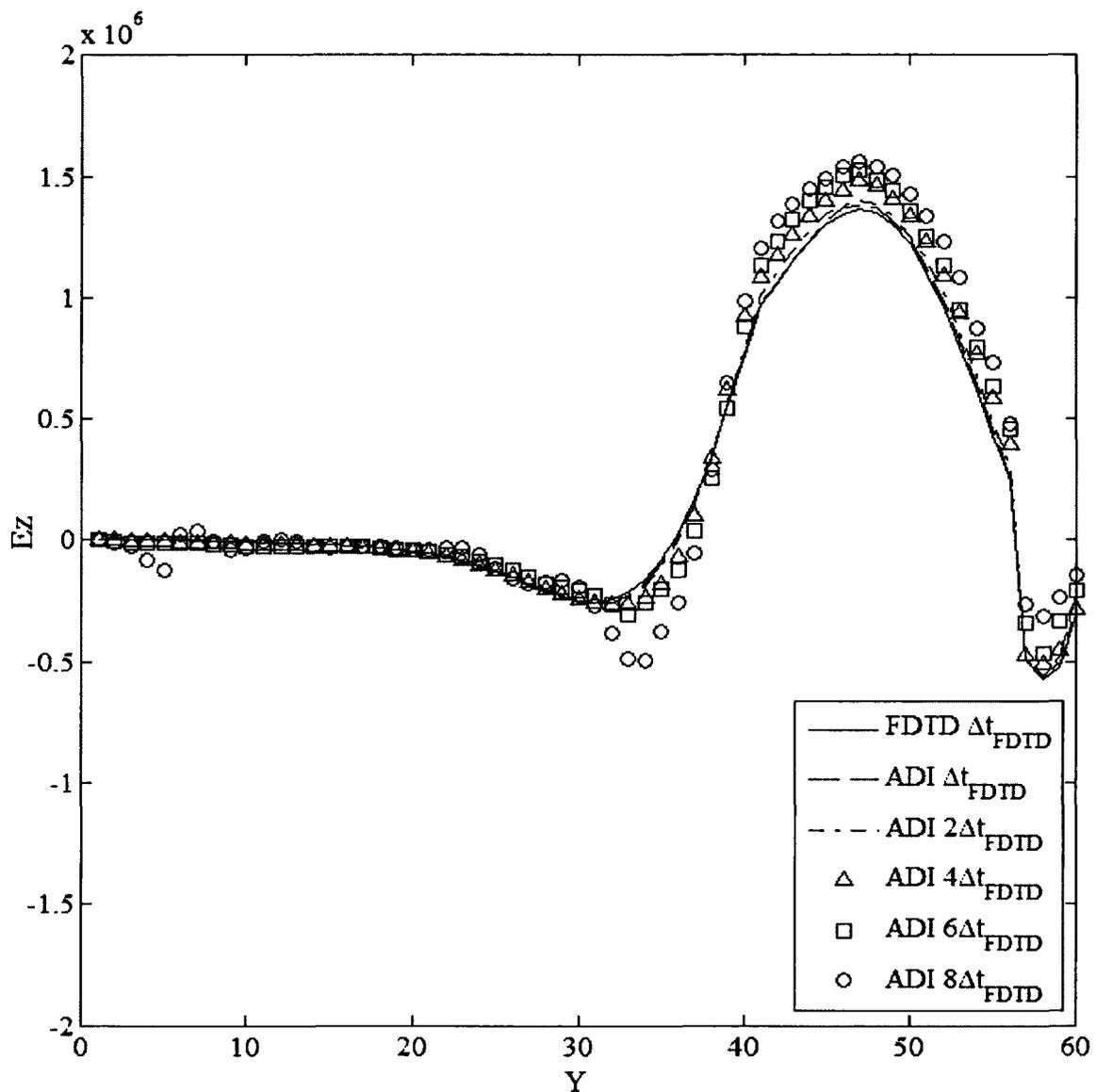


Figure 4.16e Snapshot of the values of E_z for Case 1 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $216\Delta t_{FDTD}$.

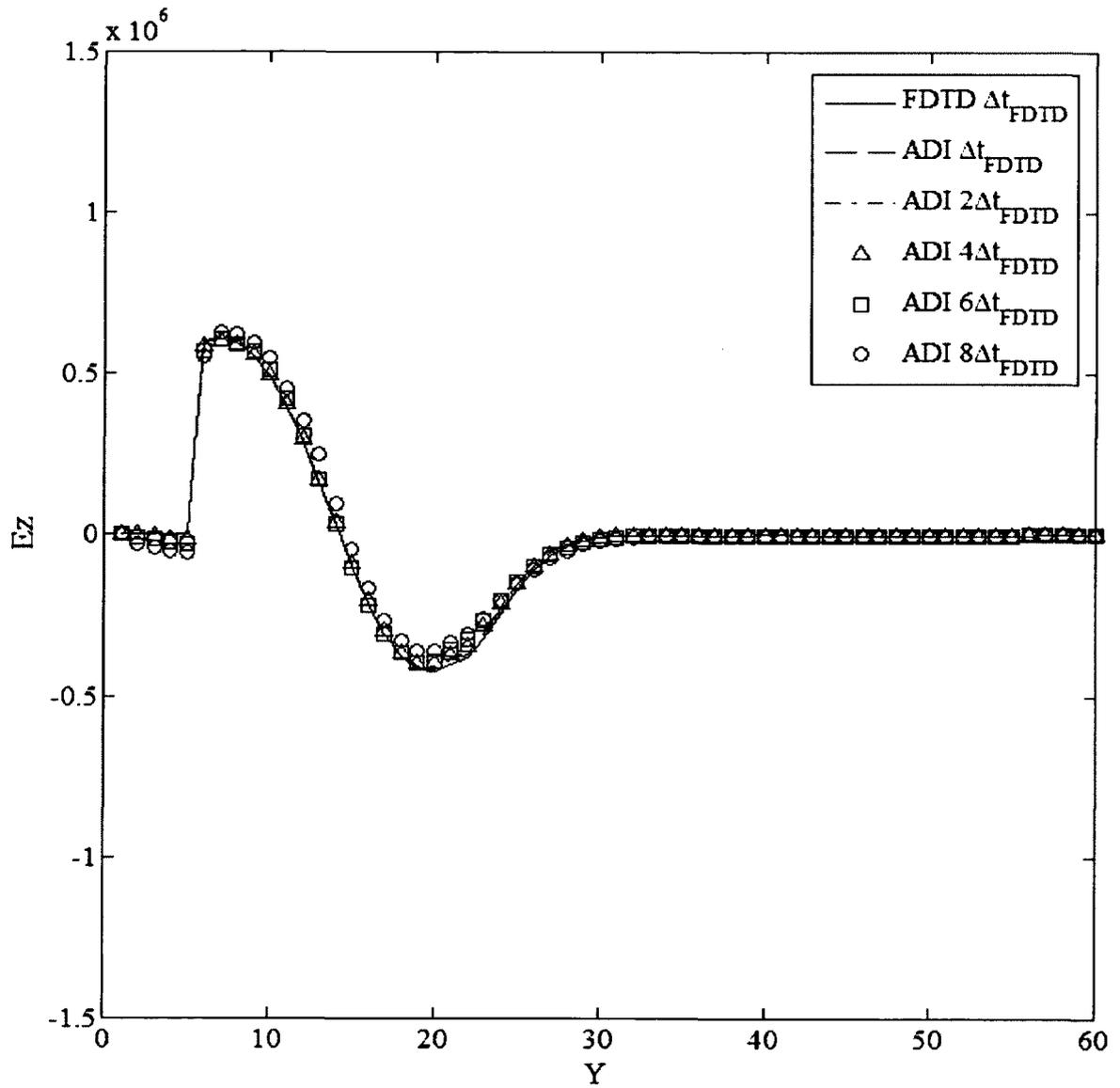


Figure 4.17a Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $120\Delta t_{FDTD}$.

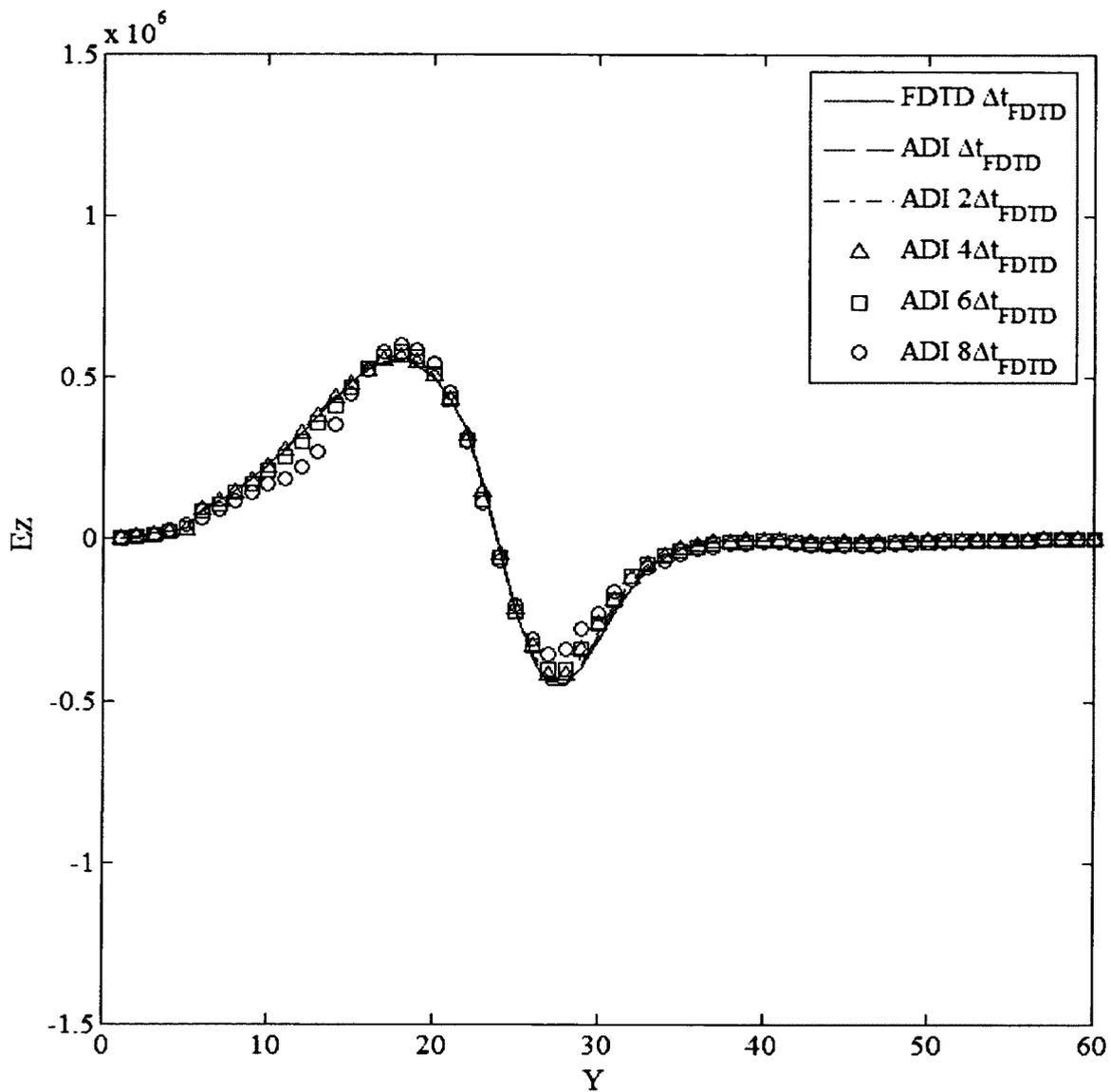


Figure 4.17b Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $144\Delta t_{FDTD}$.

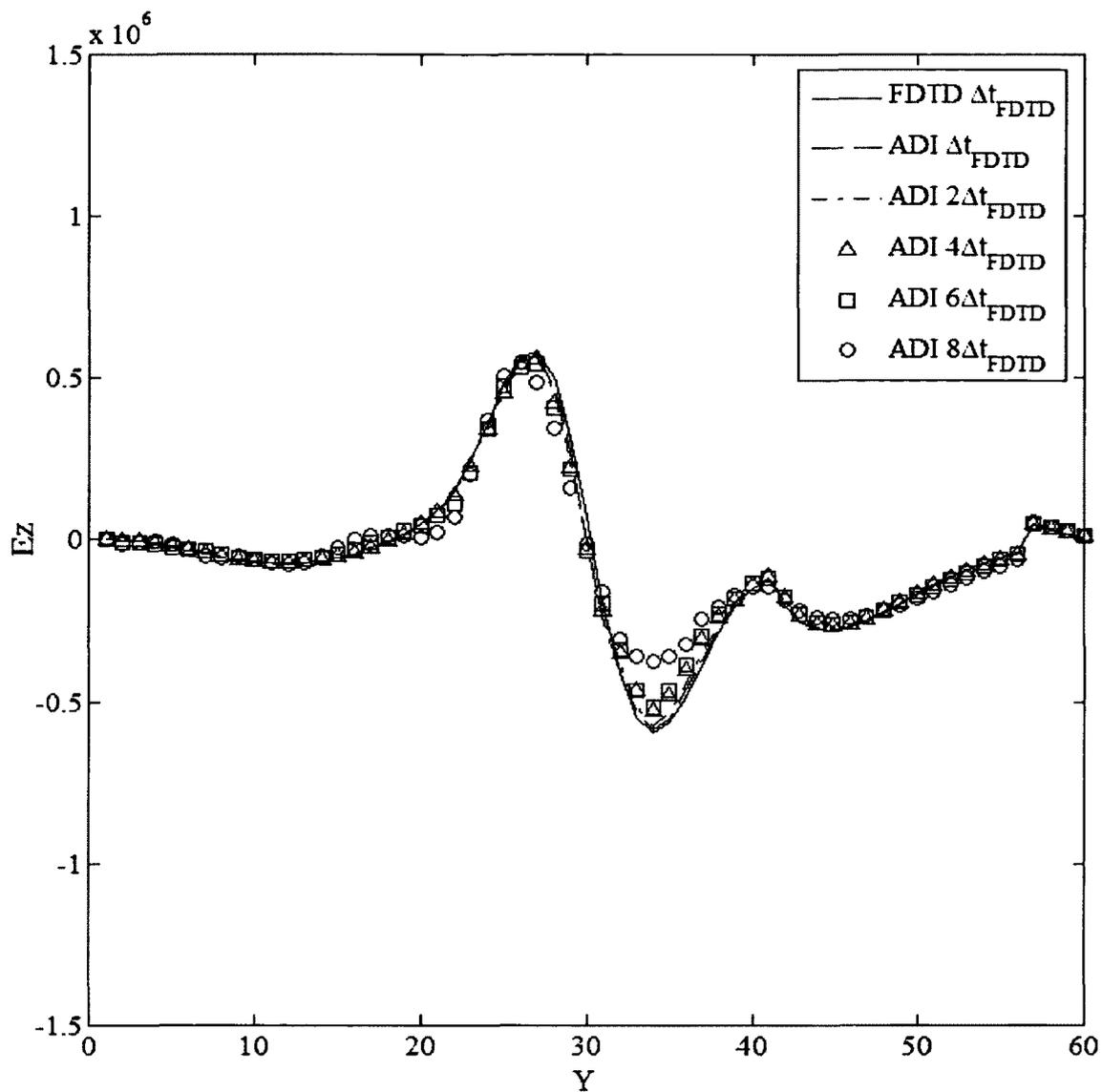


Figure 4.17c Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $168\Delta t_{FDTD}$.

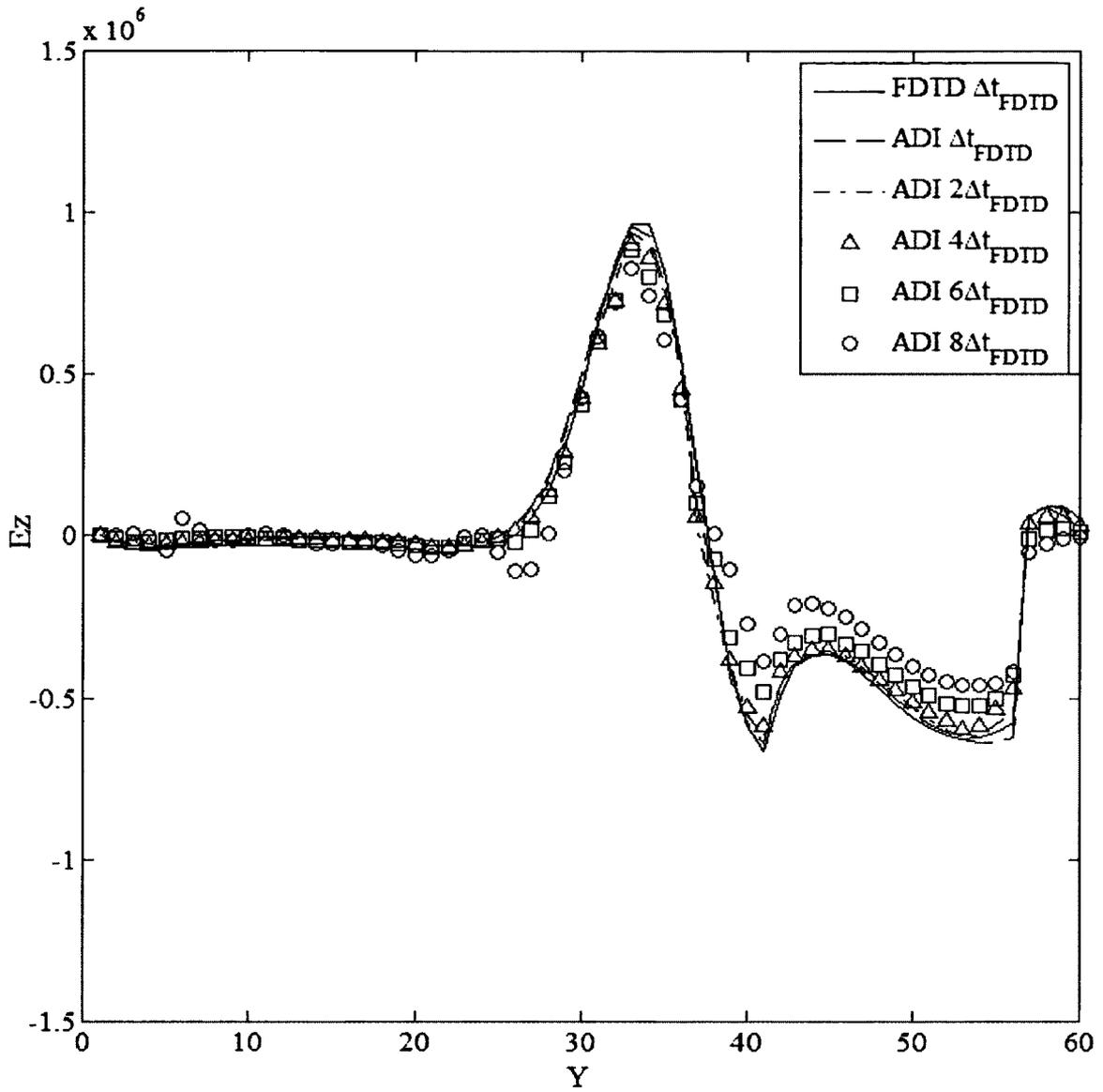


Figure 4.17d Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $192\Delta t_{FDTD}$.

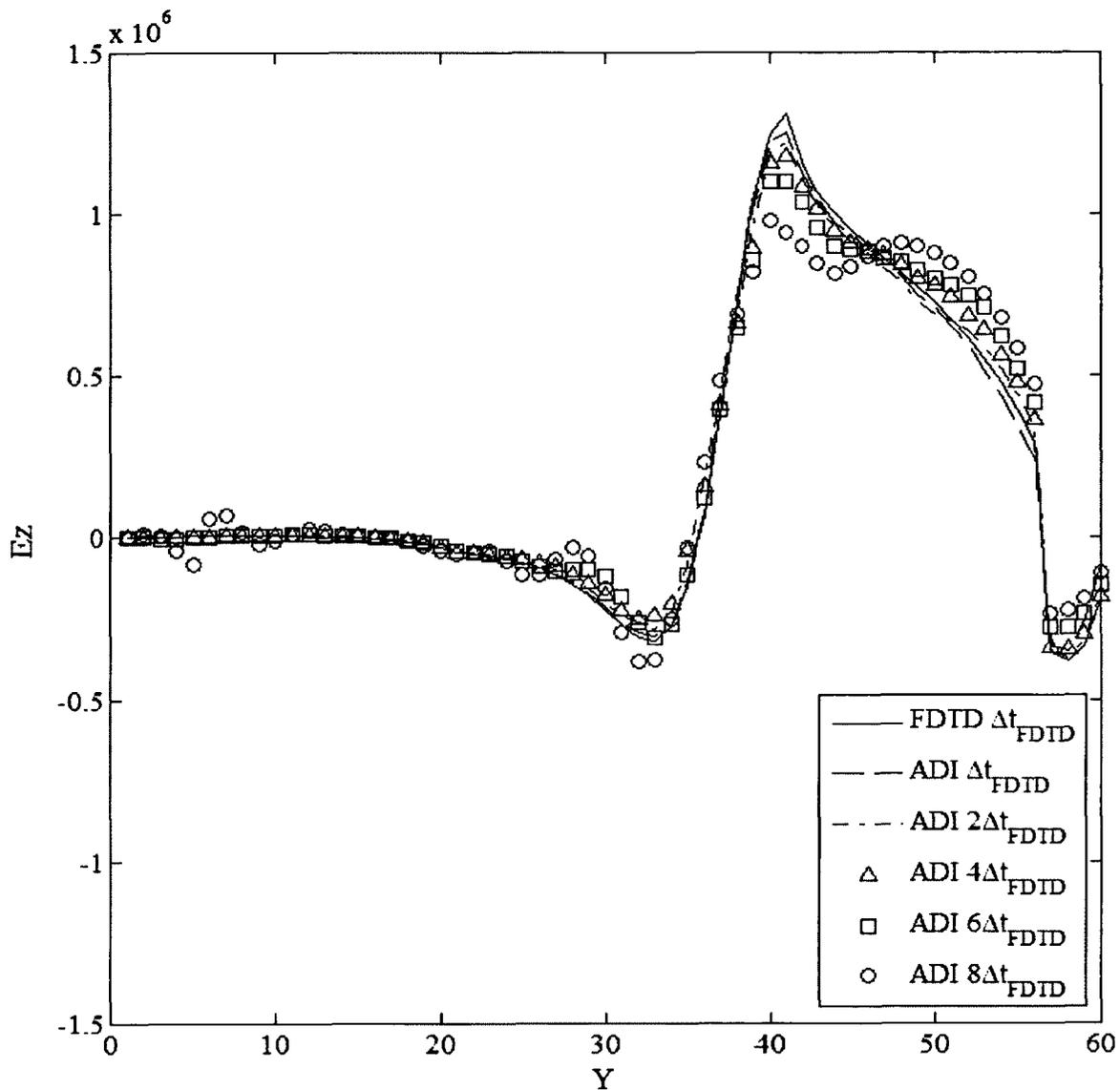


Figure 4.17e Snapshot of the values of E_z for Case 2 obtained by the conventional FDTD method and the ADI-FDTD method versus y position along the line at $x = 30 \mu\text{m}$ and $z = 30 \mu\text{m}$ after $216\Delta t_{FDTD}$.

4.2.3 Comparison of Computational Efficiency

Since the formulations of the proposed ADI-FDTD method and the conventional FDTD method are both based on the Yee's cell, two methods have the same number of field components for a same problem. As a result, the space complexities for both methods are in the same order.

Tables 4.2 and 4.3 list the running time taken by two numerical examples with different methods and different time steps. All the simulations are run using a HP laptop with a 1.87 GHz Intel Pentium Dual processor. As can be seen, if the same time step is employed in both methods, the running time taken by the proposed ADI-FDTD scheme is longer than that taken by the conventional FDTD method. This is reasonable because more calculations are involved in the proposed ADI-FDTD method at each time step.

Table 4.2 Comparison of running time for Case 1

Numerical Method and Time Steps	Number of Iterations	Running Time (s)
Convention FDTD with Δt_{FDTD}	216	66.984
ADI-FDTD with Δt_{FDTD}	216	164.335
ADI-FDTD with $2\Delta t_{FDTD}$	108	86.268
ADI-FDTD with $4\Delta t_{FDTD}$	54	43.740
ADI-FDTD with $6\Delta t_{FDTD}$	36	28.194
ADI-FDTD with $8\Delta t_{FDTD}$	27	21.409

Table 4.3 Comparison of running time for Case 2

Numerical Method and Time Steps	Number of Iterations	Running Time (s)
Convention FDTD with Δt_{FDTD}	216	63.820
ADI-FDTD with Δt_{FDTD}	216	177.792
ADI-FDTD with $2\Delta t_{FDTD}$	108	82.655
ADI-FDTD with $4\Delta t_{FDTD}$	54	39.888
ADI-FDTD with $6\Delta t_{FDTD}$	36	27.050
ADI-FDTD with $8\Delta t_{FDTD}$	27	20.131

However, since a larger time step can reduce the total number of iterations required by the simulation, the running time taken by the ADI-FDTD method decreases considerably with the time, as shown in Tables 4.2 and 4.3. In particular, the running time taken by the proposed ADI-FDTD method with $4\Delta t_{FDTD}$ is much less than that taken by the conventional FDTD method with Δt_{FDTD} .

On the one hand, using a larger time step with the proposed ADI-FDTD method reduces the running time taken by a simulation. On the other hand, it increases the numerical error as shown in the previous section. Therefore, the selection of the time step should be well-considered, depending on the requirements for different situations.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

In this dissertation, a three-dimensional ADI-FDTD scheme coupled with the Cole-Cole expression for relative dielectric coefficient of biological tissue to simulate the electromagnetic fields inside the biological tissues exposed to the nanopulses has been described. The Cole-Cole expression is approximated by the method described in [22]. The perfectly matched layer technique and the total/scattered field formulation are also employed to eliminate reflections from the boundary and to generate the plane wave, separately.

The new scheme is illustrated by numerical examples with two different biological tissues. For the purpose of comparison, both the proposed ADI-FDTD method and the conventional FDTD method are employed to run the simulations. Since the formulations of the proposed ADI-FDTD method and the conventional FDTD method are both based on the Yee's cell, the space complexities for both methods are in the same order. Numerical results show that the proposed ADI-FDTD scheme is no longer restricted by the CFL stability conditions. The new scheme provides stable solutions with larger time steps, where the conventional FDTD scheme fails. Therefore, it can greatly reduce the running time taken by the simulation. On the other hand, the truncation errors

increase with the time step. As a result, the selection of the time step depends on what kinds of problems are simulated.

Future work in this research may focus on the further improvement of stability and numerical accuracy with the ADI-FDTD method. In particular, as mentioned in the previous section, the oscillation with the larger time step is probably caused by the PML since an empirical formula for PML is employed in this research. A more suitable formula could be developed so that the reflection from boundary in simulation could be completely eliminated. In addition, the temperature distribution in the biological tissues exposed to the nanopulse could be modeled and investigated since it has more practical meanings to many biomedical applications with nanopulses.

APPENDIX A

NOMENCLATURE

E	the electric field intensity, V/m
H	the magnetic field intensity, A/m
D	the electric flux density, C/m ²
B	the magnetic flux density, Wb/m ²
J	the current density, A/m ²
ρ	the volume electric charge density, C/m ³
σ	the electrical conductivity
ε	the permittivity
ε_0	the permittivity of free space
ε_r	the relative permittivity
ε_r^*	the complex relative permittivity
μ	the permeability
μ_0	the permeability of free space
μ_r	the relative permeability
τ	relaxation time
σ_s	static conductivity
χ_e	electric susceptibility

APPENDIX B

SOURCE CODE OF EXAMPLE

```

#include "StdAfx.h"
# include <math.h>
# include <stdlib.h>
# include <stdio.h>
# include <iostream>

#define IE 61
#define JE 61
#define KE 61
#define ia 5
#define ja 5
#define ka 5
#define NFREQS 10
#define Emp 0.33

double dx[IE][JE][KE],dy[IE][JE][KE],dz[IE][JE][KE];
double ex[IE][JE][KE],ey[IE][JE][KE],ez[IE][JE][KE];
double hx[IE][JE][KE],hy[IE][JE][KE],hzz[IE][JE][KE];
double ix[IE][JE][KE],iy[IE][JE][KE],iz[IE][JE][KE];
double gax1[IE][JE][KE],gay1[IE][JE][KE],gaz1[IE][JE][KE];
double gax2[IE][JE][KE],gay2[IE][JE][KE],gaz2[IE][JE][KE];
double gax3[IE][JE][KE],gay3[IE][JE][KE],gaz3[IE][JE][KE];
double gax4[IE][JE][KE],gay4[IE][JE][KE],gaz4[IE][JE][KE];

double gbx1[IE][JE][KE],gby1[IE][JE][KE],gbz1[IE][JE][KE];
double gbx2[IE][JE][KE],gby2[IE][JE][KE],gbz2[IE][JE][KE];
double gbx3[IE][JE][KE],gby3[IE][JE][KE],gbz3[IE][JE][KE];
double gbx4[IE][JE][KE],gby4[IE][JE][KE],gbz4[IE][JE][KE];

double gdx1[IE][JE][KE],gdy1[IE][JE][KE],gdz1[IE][JE][KE];
double gdx2[IE][JE][KE],gdy2[IE][JE][KE],gdz2[IE][JE][KE];
double gdx3[IE][JE][KE],gdy3[IE][JE][KE],gdz3[IE][JE][KE];
double gdx4[IE][JE][KE],gdy4[IE][JE][KE],gdz4[IE][JE][KE];
double gcx[IE][JE][KE],gcy[IE][JE][KE],gcz[IE][JE][KE];
double gx[IE][JE][KE],gy[IE][JE][KE],gz[IE][JE][KE];
double sx1[IE][JE][KE][3],sy1[IE][JE][KE][3],sz1[IE][JE][KE][3];
double sx2[IE][JE][KE][3],sy2[IE][JE][KE][3],sz2[IE][JE][KE][3];
double sx3[IE][JE][KE][3],sy3[IE][JE][KE][3],sz3[IE][JE][KE][3];
double sx4[IE][JE][KE][3],sy4[IE][JE][KE][3],sz4[IE][JE][KE][3];

double idxl[ia][JE][KE],idxh[ia][JE][KE];
double ihxl[ia][JE][KE],ihxh[ia][JE][KE];
double idyl[IE][ja][KE],idyh[IE][ja][KE];
double ihyl[IE][ja][KE],ihyh[IE][ja][KE];
double idzl[IE][JE][ka],idzh[IE][JE][ka];
double ihzl[IE][JE][ka],ihzh[IE][JE][ka];

double curl_hx[IE][JE][KE],curl_hy[IE][JE][KE],curl_hz[IE][JE][KE];
double curl_ex[IE][JE][KE],curl_ey[IE][JE][KE],curl_ez[IE][JE][KE];
double ex_temp[IE][JE][KE],ey_temp[IE][JE][KE],ez_temp[IE][JE][KE];

double inc_gaz1[JE];
double inc_gaz2[JE];
double inc_gaz3[JE];
double inc_gaz4[JE];
double inc_gbz1[JE];

```

```

double inc_gbz2[JE];
double inc_gbz3[JE];
double inc_gbz4[JE];

double inc_gdz1[JE];
double inc_gdz2[JE];
double inc_gdz3[JE];
double inc_gdz4[JE];
double inc_gcz[JE];
double inc_gz[JE];
double inc_sz1[JE][3];
double inc_sz2[JE][3];
double inc_sz3[JE][3];
double inc_sz4[JE][3];

int main(int argc, char* argv[]){

int n,i,j,k,ic,jc,kc,nsteps,n_pml;
double ddx,c0,dt,epsz,muz,pi,npml,T,time_scale,cdt_2,cdt_2dx,cdt_2dx_sq;
int ib,jb,kb,numsph;
int NCUR,NPR2,NPR1;
double xn,xxn,curl_e;
double t0,spread,pulse;
double ez_inc[JE],ez_inc_temp[JE],hx_inc[JE],hx_inc_temp[JE],dz_inc[JE],iz_inc[JE];
double inc_iTA4[JE];

double ez_low_m1,ez_low_m2,ez_high_m1,ez_high_m2;
clock_t start, finish;
double duration;
int ixh, jyh, kzh;

double gi1[IE],gi2[IE],gi3[IE];
double gj1[JE],gj2[JE],gj3[JE];
double gk1[KE],gk2[KE],gk3[KE];
double fi1[IE],fi2[IE],fi3[IE];
double fj1[JE],fj2[JE],fj3[JE];
double fk1[KE],fk2[KE],fk3[KE];

double radius[10],epsilon[10],sigma[10],eps,cond;
double dell1[10],dell2[10],dell3[10],dell4[10];
double tau1[10],tau2[10],tau3[10],tau4[10];
double alpha1[10],alpha2[10],alpha3[10],alpha4[10];
double A1,A2,A3,A4,B1,B2,B3,B4,C1,C2,C3,C4;
double dist,xdist,ydist,zdist,curl_h;

double TA1[IE],TA2[IE],TA3[IE],TA4[IE];
double TAf[IE],TAe[IE];
double iTA4[IE][JE][KE];

FILE *fp4,*fp5,*fp6,*fp7,*fp8;
ic = (IE-1)/2 ;
jc = (JE-1)/2 ;
kc = (KE-1)/2 ;
ib = IE - ia - 1;
jb = JE - ja - 1;
kb = KE - ka - 1;

```

```

pi = 3.14159;
epsz = 8.8e-12;
muz = 4*pi*1.e-7;
ddx = 60e-6/(1E-1);
c0 = 3e8;
time_scale=1;
dt = (ddx/6e8)*time_scale;
cdt_2=c0*dt/2;
cdt_2dx=c0*dt/2/ddx;
cdt_2dx_sq=cdt_2dx*cdt_2dx;

NCUR=2;
NPR1=1;
NPR2=0;

/* Initialize the arrays */
for ( j=0; j < JE; j++ ) {
    for ( k=0; k < KE; k++ ) {
        for ( i=0; i < IE; i++ ) {
            ex[i][j][k]= 0.0 ;
            ey[i][j][k]= 0.0 ;
            ez[i][j][k]= 0.0 ;
            dx[i][j][k]= 0.0 ;
            dy[i][j][k]= 0.0 ;
            dz[i][j][k]= 0.0 ;
            hx[i][j][k]= 0.0 ;
            hy[i][j][k]= 0.0 ;
            hzz[i][j][k]= 0.0 ;
            ix[i][j][k]= 0.0 ;
            iy[i][j][k]= 0.0 ;
            iz[i][j][k]= 0.0 ;

            gcx[i][j][k]= 1.0;
            gcy[i][j][k]= 1.0;
            gcz[i][j][k]= 1.0;
            gx[i][j][k]=0.0;
            gy[i][j][k]=0.0;
            gz[i][j][k]=0.0;

            gax1[i][j][k]=0.0;
            gax2[i][j][k]=0.0;
            gax3[i][j][k]=0.0;
            gax4[i][j][k]=0.0;
            gay1[i][j][k]=0.0;
            gay2[i][j][k]=0.0;
            gay3[i][j][k]=0.0;
            gay4[i][j][k]=0.0;
            gaz1[i][j][k]=0.0;
            gaz2[i][j][k]=0.0;
            gaz3[i][j][k]=0.0;
            gaz4[i][j][k]=0.0;

            gbx1[i][j][k]=0.0;
            gbx2[i][j][k]=0.0;
            gbx3[i][j][k]=0.0;
            gbx4[i][j][k]=0.0;

```

```

gby1[i][j][k]=0.0;
gby2[i][j][k]=0.0;
gby3[i][j][k]=0.0;
gby4[i][j][k]=0.0;
gbz1[i][j][k]=0.0;
gbz2[i][j][k]=0.0;
gbz3[i][j][k]=0.0;
gbz4[i][j][k]=0.0;

gdx1[i][j][k]=0.0;
gdx2[i][j][k]=0.0;
gdx3[i][j][k]=0.0;
gdx4[i][j][k]=0.0;
gdy1[i][j][k]=0.0;
gdy2[i][j][k]=0.0;
gdy3[i][j][k]=0.0;
gdy4[i][j][k]=0.0;
gdz1[i][j][k]=0.0;
gdz2[i][j][k]=0.0;
gdz3[i][j][k]=0.0;
gdz4[i][j][k]=0.0;
} } }

//////////

for ( j=0; j < JE; j++ ) {
  for ( k=0; k < KE; k++ ) {
    for ( i=0; i < IE; i++ ) {
      for(int temp=0;temp<3;temp++){
        sx1[i][j][k][temp]=0.0;
        sx2[i][j][k][temp]=0.0;
        sx3[i][j][k][temp]=0.0;
        sx4[i][j][k][temp]=0.0;
        sy1[i][j][k][temp]=0.0;
        sy2[i][j][k][temp]=0.0;
        sy3[i][j][k][temp]=0.0;
        sy4[i][j][k][temp]=0.0;
        sz1[i][j][k][temp]=0.0;
        sz2[i][j][k][temp]=0.0;
        sz3[i][j][k][temp]=0.0;
        sz4[i][j][k][temp]=0.0;
      } } }

      for(j=0;j<JE;j++){
        ez_inc[j]=0;
        ez_inc_temp[j]=0;
        dz_inc[j]=0;
        hx_inc[j]=0;
        hx_inc_temp[j]=0;
        iz_inc[j]=0;
        inc_gcz[j]= 1.0;
        inc_gz[j]=0.0;
        inc_gaz1[j]=0.0;
        inc_gaz2[j]=0.0;
        inc_gaz3[j]=0.0;
        inc_gaz4[j]=0.0;
      } } }

```

```

        inc_gbz1[j]=0.0;
        inc_gbz2[j]=0.0;
        inc_gbz3[j]=0.0;
        inc_gbz4[j]=0.0;
        inc_gdz1[j]=0.0;
        inc_gdz2[j]=0.0;
        inc_gdz3[j]=0.0;
        inc_gdz4[j]=0.0;
        inc_iTA4[j]=0.0;
    }

    ez_low_m1=0;
    ez_low_m2=0;
    ez_high_m1=0;
    ez_high_m2=0;

    ////

    for ( i=0; i < ia; i++ ) {
        for ( j=0; j < JE; j++ ) {
            for ( k=0; k < KE; k++ ) {
                idxl[i][j][k] = 0.0;
                idxh[i][j][k] = 0.0;
                ihxl[i][j][k] = 0.0;
                ihxh[i][j][k] = 0.0;
            } } }

    for ( i=0; i < IE; i++ ) {
        for ( j=0; j < ja; j++ ) {
            for ( k=0; k < KE; k++ ) {
                idyl[i][j][k] = 0.0;
                idyh[i][j][k] = 0.0;
                ihyl[i][j][k] = 0.0;
                ihyh[i][j][k] = 0.0;
            } } }

    for ( i=0; i < IE; i++ ) {
        for ( j=0; j < JE; j++ ) {
            for ( k=0; k < ka; k++ ) {
                idzl[i][j][k] = 0.0;
                idzh[i][j][k] = 0.0;
                ihzl[i][j][k] = 0.0;
                ihzh[i][j][k] = 0.0;
            } } }

    for ( i=0; i < IE; i++){
        for ( j=0; j < JE; j++){
            for ( k=0; k < KE; k++){
                curl_hx[i][j][k]=0.0;
                curl_hy[i][j][k]=0.0;
                curl_hz[i][j][k]=0.0;
                curl_ex[i][j][k]=0.0;
                curl_ey[i][j][k]=0.0;
                curl_ez[i][j][k]=0.0;
                ex_temp[i][j][k]=0.0;
                ey_temp[i][j][k]=0.0;
            } } }
    }

```

```

                ez_temp[i][j][k]=0.0;
        } } }

        /* Boundary Conditions */

        for ( i=0; i < IE; i++ ) {
                gi1[i] = 0.;
                fi1[i] = 0.;
                gi2[i] = 1.;
                fi2[i] = 1.;
                gi3[i] = 1.;
                fi3[i] = 1.;
        }

        for ( j=0; j < JE; j++ ) {
                gj1[j] = 0.;
                fj1[j] = 0.;
                gj2[j] = 1.;
                fj2[j] = 1.;
                gj3[j] = 1.;
                fj3[j] = 1.;
        }

        for ( k=0; k < IE; k++ ) {
                gk1[k] = 0.;
                fk1[k] = 0.;
                gk2[k] = 1.;
                fk2[k] = 1.;
                gk3[k] = 1.;
                fk3[k] = 1.;
        }

        npml=5;
        n_pml=(int) npml;

        for ( i=0; i < n_pml; i++ ) {
                xxn = (npml-i)/npml;
                xn = Emp*pow(xxn,3.);
                fi1[i] = 2*xn;
                fi1[IE-i-1] = 2*xn;
                gi2[i] = 1./(1.+xn);
                gi2[IE-i-1] = 1./(1.+xn);
                gi3[i] = (1.-xn)/(1.+xn);
                gi3[IE-i-1] = (1.-xn)/(1.+xn);
                xxn = (npml-i-.5)/npml;
                xn = Emp*pow(xxn,3.);
                gi1[i] =2*xn;
                gi1[IE-i-2] = 2*xn;
                fi2[i] = 1./(1.+xn);
                fi2[IE-i-2] = 1./(1.+xn);
                fi3[i] = (1.-xn)/(1.+xn);
                fi3[IE-i-2] = (1.-xn)/(1.+xn);
        }

        for ( j=0; j < n_pml; j++ ) {
                xxn = (npml-j)/npml;

```

```

    xn = Emp*pow(xxn,3.);
    fj1[j] = 2*xn;
    fj1[JE-j-1] = 2*xn;
    gj2[j] = 1./(1.+xn);
    gj2[JE-j-1] = 1./(1.+xn);
    gj3[j] = (1.-xn)/(1.+xn);
    gj3[JE-j-1] = (1.-xn)/(1.+xn);
    xxn = (npml-j-.5)/npml;
    xn = Emp*pow(xxn,3.);
    gj1[j] = 2*xn;
    gj1[JE-j-2] = 2*xn;
    fj2[j] = 1./(1.+xn);
    fj2[JE-j-2] = 1./(1.+xn);
    fj3[j] = (1.-xn)/(1.+xn);
    fj3[JE-j-2] = (1.-xn)/(1.+xn);
}

for ( k=0; k < n_pml; k++ ) {
    xxn = (npml-k)/npml;
    xn = Emp*pow(xxn,3.);
    fk1[k] = 2*xn;
    fk1[KE-k-1] = 2*xn;
    gk2[k] = 1./(1.+xn);
    gk2[KE-k-1] = 1./(1.+xn);
    gk3[k] = (1.-xn)/(1.+xn);
    gk3[KE-k-1] = (1.-xn)/(1.+xn);
    xxn = (npml-k-.5)/npml;
    xn = Emp*pow(xxn,3.);
    gk1[k] = 2*xn;
    gk1[KE-k-2] = 2*xn;
    fk2[k] = 1./(1.+xn);
    fk2[KE-k-2] = 1./(1.+xn);
    fk3[k] = (1.-xn)/(1.+xn);
    fk3[KE-k-2] = (1.-xn)/(1.+xn);
}

for( i=0; i < IE; i++ ) {
    TA1[i]=0.;
    TA2[i]=0.;
    TA3[i]=0.;
    TA4[i]=0.;
    TAf[i]=0.;
    TAe[i]=0.;
}

for ( i=0; i < IE; i++ ){
    for ( j=0; j < JE; j++ ){
        for ( k=0; k < KE; k++ ){
            iTA4[i][j][k]=0.0;
        } } }

/* Specify the dielectric sphere */
epsilon[0] = 1.;
sigma[0] = 0.;
dell1[0]=0.;
dell2[0]=0.;

```

```

dell3[0]=0.;
dell4[0]=0.;
tau1[0]=0.;
tau2[0]=0.;
tau3[0]=0.;
tau4[0]=0.;
alpha1[0]=0.;
alpha2[0]=0.;
alpha3[0]=0.;
alpha4[0]=0.;

radius[1]=10;
epsilon[1]=2.5;
sigma[1]=0.01;
dell1[1]=3.;
dell2[1]=15.;
dell3[1]=5.0e4;
dell4[1]=5.0e7;
tau1[1]=17.680e-12;
tau2[1]=63.660e-9;
tau3[1]=454.700e-6;
tau4[1]=13.260e-3;
alpha1[1]=0.1;
alpha2[1]=0.1;
alpha3[1]=0.1;
alpha4[1]=0.;
numsph=1;

/* Calculate gax,gbx */
for ( i = 0; i < IE; i++ ) {
    for ( j = 0; j < JE; j++ ) {
        for ( k = 0; k < KE; k++ ) {
            eps = epsilon[0];
            cond = sigma[0];
            C1=pow(tau1[0]/dt/2,1-alpha1[0]);
            C2=pow(tau2[0]/dt/2,1-alpha2[0]);
            C3=pow(tau3[0]/dt/2,1-alpha3[0]);
            C4=pow(tau4[0]/dt/2,1-alpha4[0]);
            B1=alpha1[0];
            B2=alpha2[0];
            B3=alpha3[0];
            B4=alpha4[0];
            A1=dell1[0]/(1+C1);
            A2=dell2[0]/(1+C2);
            A3=dell3[0]/(1+C3);
            A4=dell4[0]/(1+C4);

            ydist = (jc-j);
            xdist = (ic-i-.5);
            zdist = (kc-k);
            xdist = sqrt(pow(xdist,2.));
            ydist = sqrt(pow(ydist,2.));
            zdist = sqrt(pow(zdist,2.));

            for (n=1; n<= numsph; n++) {
                if((xdist<= radius[n])&&(ydist<= radius[n])&&(zdist<= radius[n])) {

```

```

        eps = epsilon[n];
        cond = sigma[n] ;
        C1=pow(tau1[n]/dt/2,1-alpha1[n]);
        C2=pow(tau2[n]/dt/2,1-alpha2[n]);
        C3=pow(tau3[n]/dt/2,1-alpha3[n]);
        C4=pow(tau4[n]/dt/2,1-alpha4[n]);
        B1=alpha1[n];
        B2=alpha2[n];
        B3=alpha3[n];
        B4=alpha4[n];
        A1=dell1[n]/(1+C1);
        A2=dell2[n]/(1+C2);
        A3=dell3[n]/(1+C3);
        A4=dell4[n]/(1+C4);
    } }

    gcx[i][j][k]=1./(eps+(cond*dt/2/epsz)+A1+A2+A3+A4);
    gax1[i][j][k]=(1-B1)*C1/(1+C1);
    gax2[i][j][k]=(1-B2)*C2/(1+C2);
    gax3[i][j][k]=(1-B3)*C3/(1+C3);
    gax4[i][j][k]=(1-B4)*C4/(1+C4);
    gbx1[i][j][k]=0.5*(1-B1)*B1*C1/(1+C1);
    gbx2[i][j][k]=0.5*(1-B2)*B2*C2/(1+C2);
    gbx3[i][j][k]=0.5*(1-B3)*B3*C3/(1+C3);
    gbx4[i][j][k]=0.5*(1-B4)*B4*C4/(1+C4);
    gdx1[i][j][k]=A1;
    gdx2[i][j][k]=A2;
    gdx3[i][j][k]=A3;
    gdx4[i][j][k]=A4;
    gx[i][j][k]=cond*dt/2/epsz;
} } }

/* Calculate gay,gby */
for ( i = 0; i < IE; i++ ) {
    for ( j = 0; j < JE; j++ ) {
        for ( k = 0; k < KE; k++ ) {
            eps = epsilon[0];
            cond = sigma[0];
            C1=pow(tau1[0]/dt/2,1-alpha1[0]);
            C2=pow(tau2[0]/dt/2,1-alpha2[0]);
            C3=pow(tau3[0]/dt/2,1-alpha3[0]);
            C4=pow(tau4[0]/dt/2,1-alpha4[0]);
            B1=alpha1[0];
            B2=alpha2[0];
            B3=alpha3[0];
            B4=alpha4[0];
            A1=dell1[0]/(1+C1);
            A2=dell2[0]/(1+C2);
            A3=dell3[0]/(1+C3);
            A4=dell4[0]/(1+C4);
            xdist = (ic-i);
            ydist = (jc-j-.5);
            zdist = (kc-k);
            xdist = sqrt(pow(xdist,2.));
            ydist = sqrt(pow(ydist,2.));
            zdist = sqrt(pow(zdist,2.));

```

```

for (n=1; n<= numsph; n++) {
    if((xdist<= radius[n])&&(ydist<= radius[n])&&(zdist<= radius[n])) {
        eps = epsilon[n] ;
        cond = sigma[n] ;
        C1=pow(tau1[n]/dt/2,1-alpha1[n]);
        C2=pow(tau2[n]/dt/2,1-alpha2[n]);
        C3=pow(tau3[n]/dt/2,1-alpha3[n]);
        C4=pow(tau4[n]/dt/2,1-alpha4[n]);
        B1=alpha1[n];
        B2=alpha2[n];
        B3=alpha3[n];
        B4=alpha4[n];
        A1=dell1[n]/(1+C1);
        A2=dell2[n]/(1+C2);
        A3=dell3[n]/(1+C3);
        A4=dell4[n]/(1+C4);
    }
}

gcy[i][j][k]=1./(eps+(cond*dt/2/epsz)+A1+A2+A3+A4);
gay1[i][j][k]=(1-B1)*C1/(1+C1);
gay2[i][j][k]=(1-B2)*C2/(1+C2);
gay3[i][j][k]=(1-B3)*C3/(1+C3);
gay4[i][j][k]=(1-B4)*C4/(1+C4);
gby1[i][j][k]=0.5*(1-B1)*B1*C1/(1+C1);
gby2[i][j][k]=0.5*(1-B2)*B2*C2/(1+C2);
gby3[i][j][k]=0.5*(1-B3)*B3*C3/(1+C3);
gby4[i][j][k]=0.5*(1-B4)*B4*C4/(1+C4);
gdy1[i][j][k]=A1;
gdy2[i][j][k]=A2;
gdy3[i][j][k]=A3;
gdy4[i][j][k]=A4;
gy[i][j][k]=cond*dt/2/epsz;
}}}

/* Calculate gaz,gbz */
for ( i = 0; i < IE; i++ ) {
    for ( j = 0; j < JE; j++ ) {
        for ( k = 0; k < KE; k++ ) {
            eps = epsilon[0];
            cond = sigma[0];
            C1=pow(tau1[0]/dt/2,1-alpha1[0]);
            C2=pow(tau2[0]/dt/2,1-alpha2[0]);
            C3=pow(tau3[0]/dt/2,1-alpha3[0]);
            C4=pow(tau4[0]/dt/2,1-alpha4[0]);
            B1=alpha1[0];
            B2=alpha2[0];
            B3=alpha3[0];
            B4=alpha4[0];
            A1=dell1[0]/(1+C1);
            A2=dell2[0]/(1+C2);
            A3=dell3[0]/(1+C3);
            A4=dell4[0]/(1+C4);
            xdist = (ic-i);
            ydist = (jc-j);
            zdist = (kc-k-.5);
            xdist = sqrt(pow(xdist,2.));

```

```

ydist = sqrt(pow(ydist,2.));
zdist = sqrt(pow(zdist,2.));
for (n=1; n<= numsph; n++) {
    if((xdist<= radius[n])&&(ydist<= radius[n])&&(zdist<= radius[n])) {
        eps = epsilon[n] ;
        cond = sigma[n];
        C1=pow(tau1[n]/dt/2,1-alpha1[n]);
        C2=pow(tau2[n]/dt/2,1-alpha2[n]);
        C3=pow(tau3[n]/dt/2,1-alpha3[n]);
        C4=pow(tau4[n]/dt/2,1-alpha4[n]);
        B1=alpha1[n];
        B2=alpha2[n];
        B3=alpha3[n];
        B4=alpha4[n];
        A1=dell1[n]/(1+C1);
        A2=dell2[n]/(1+C2);
        A3=dell3[n]/(1+C3);
        A4=dell4[n]/(1+C4);
    } }

gcz[i][j][k]=1./(eps+(cond*dt/2/epsz)+A1+A2+A3+A4);
gaz1[i][j][k]=(1-B1)*C1/(1+C1);
gaz2[i][j][k]=(1-B2)*C2/(1+C2);
gaz3[i][j][k]=(1-B3)*C3/(1+C3);
gaz4[i][j][k]=(1-B4)*C4/(1+C4);
gbz1[i][j][k]=0.5*(1-B1)*B1*C1/(1+C1);
gbz2[i][j][k]=0.5*(1-B2)*B2*C2/(1+C2);
gbz3[i][j][k]=0.5*(1-B3)*B3*C3/(1+C3);
gbz4[i][j][k]=0.5*(1-B4)*B4*C4/(1+C4);
gdz1[i][j][k]=A1;
gdz2[i][j][k]=A2;
gdz3[i][j][k]=A3;
gdz4[i][j][k]=A4;
gz[i][j][k]=cond*dt/2/epsz;
}}}

t0 = 40.0;
spread = 10.0;
T=0;
nsteps = 2;

fp4 = fopen("120.txt","w");
fprintf(fp4,"TITLE=ELECTRIC FIELD\n");
fprintf(fp4,"VARIABLES=X, Y, Ez\n");
fp5 = fopen("144.txt","w");
fprintf(fp5,"TITLE=ELECTRIC FIELD\n");
fprintf(fp5,"VARIABLES=X, Y, Ez\n");
fp6 = fopen("168.txt","w");
fprintf(fp6,"TITLE=ELECTRIC FIELD\n");
fprintf(fp6,"VARIABLES= X, Y, Ez\n");
fp7 = fopen("192.txt","w");
fprintf(fp7,"TITLE=ELECTRIC FIELD\n");
fprintf(fp7,"VARIABLES= X, Y, Ez\n");
fp8 = fopen("216.txt","w");
fprintf(fp8,"TITLE=ELECTRIC FIELD\n");
fprintf(fp8,"VARIABLES= X, Y, Ez\n");

```

```

nsteps=220;
for ( n=1; n <=nsteps ; n++) {

    T = T + 0.5;
    NPR2=NPR1;
    NPR1=NCUR;
    NCUR=(NCUR+1)%3;
    /* ---- Start of the Main FDTD loop ---- */

    /* Calculate the incident buffer */
    for ( j=1; j < JE; j++) {
        dz_inc[j] =dz_inc[j]+cdt_2dx*( hx_inc[j-1]- hx_inc[j]);
    }

    pulse=-1.0e6*(100.-T)/sqrt(200.)*exp(-(T-100.)*(T-100.)/400.0);
    dz_inc[3]=pulse;

    for ( j=0; j < JE; j++) {
        ez_inc_temp[j]=ez_inc[j];
    }

    for ( j=1; j < JE-1; j++) {
        B1=inc_gaz1[j]*inc_sz1[j][NPR1]+inc_gbz1[j]*inc_sz1[j][NPR2];
        B2=inc_gaz2[j]*inc_sz2[j][NPR1]+inc_gbz2[j]*inc_sz2[j][NPR2];
        B3=inc_gaz3[j]*inc_sz3[j][NPR1]+inc_gbz3[j]*inc_sz3[j][NPR2];
        B4=inc_gaz4[j]*inc_sz4[j][NPR1]+inc_gbz4[j]*inc_sz4[j][NPR2];
        ez_inc[j] = inc_gcz[j]*(dz_inc[j] - iz_inc[j]-B1-B2-B3-B4);
        iz_inc[j]= iz_inc[j] + inc_gz[j]*ez_inc[j];
        inc_sz1[j][NCUR]=B1+inc_gdz1[j]*ez_inc[j];
        inc_sz2[j][NCUR]=B2+inc_gdz2[j]*ez_inc[j];
        inc_sz3[j][NCUR]=B3+inc_gdz3[j]*ez_inc[j];
        inc_sz4[j][NCUR]=B4+inc_gdz4[j]*ez_inc[j];
    }

    /* Boundary conditions for the incident buffer*/
    ez_inc[0] = ez_low_m2;
    ez_low_m2 = ez_low_m1;
    ez_low_m1 = ez_inc[1];

    ez_inc[JE-1] = ez_high_m2;
    ez_high_m2 = ez_high_m1;
    ez_high_m1 = ez_inc[JE-2];

    /* Calculate the incident field */
    for ( j=0; j < JE; j++) {
        hx_inc_temp[j]=hx_inc[j];
    }

    for ( j=0; j < JE-1; j++) {
        hx_inc[j] = hx_inc[j] +cdt_2dx*( ez_inc_temp[j] - ez_inc_temp[j+1] );
    }

    /* Calculate the Dx field */
    for ( i=1; i < IE-2; i++) {
        for ( j=1; j < JE-1; j++) {

```

```

        for ( k=1; k < KE-1; k++ ) {
            B1=gax1[i][j][k]*sx1[i][j][k][NPR1]+gbx1[i][j][k]*sx1[i][j][k][NPR2];
            B2=gax2[i][j][k]*sx2[i][j][k][NPR1]+gbx2[i][j][k]*sx2[i][j][k][NPR2];
            B3=gax3[i][j][k]*sx3[i][j][k][NPR1]+gbx3[i][j][k]*sx3[i][j][k][NPR2];
            B4=gax4[i][j][k]*sx4[i][j][k][NPR1]+gbx4[i][j][k]*sx4[i][j][k][NPR2];
            iTA4[i][j][k] = gcx[i][j][k]*(- ix[i][j][k]-B1-B2-B3-B4);
        }
    }

    for ( i=1; i < IE-2; i++ ) {
        for ( k=1; k < KE-1; k++ ) {
            for ( j=1; j < JE-1; j++ ) {
                TA1[j]=cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fi2[i]*fj2[j-1]*(1+fk1[k])*gcx[i][j-1][k];
                TA2[j]=1+cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fi2[i]*(fj2[j-1]+fj2[j])*(1+fk1[k])*gcx[i][j][k];
                TA3[j]=cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fi2[i]*fj2[j]*(1+fk1[k])*gcx[i][j+1][k];
                curl_h = ( hzz[i][j][k] - hzz[i][j-1][k] - hy[i][j][k] + hy[i][j][k-1])/ddx ;
                curl_hx[i][j][k]=curl_hx[i][j][k]+curl_h;
                curl_e = ( ex[i][j+1][k] - ex[i][j][k] - ey[i+1][j][k] + ey[i][j][k])/ddx ;
                curl_ez[i][j][k]=curl_ez[i][j][k]+curl_e;
                TA4[j]=gj3[j]*gk3[k]*dx[i][j][k]-cdt_2dx*gj2[j]*gk2[k]*(hy[i][j][k]-hy[i][j][k-1])
                    +gj2[j]*gk2[k]*gi1[i]*cdt_2*curl_hx[i][j][k]+cdt_2dx*gj2[j]*gk2[k]*(1+gi1[i])
                    *(fi3[i]*fj3[j]*hzz[i][j][k]-cdt_2dx*fi2[i]*fj2[j]*(ey[i+1][j][k]-ey[i][j][k])
                    +fi2[i]*fj2[j]*fk1[k]*cdt_2*curl_ez[i][j][k]-fi3[i]*fj3[j-1]*hzz[i][j-1][k]
                    +cdt_2dx*fi2[i]*fj2[j-1]*(ey[i+1][j-1][k]-ey[i][j-1][k])-fi2[i]*fj2[j-1]
                    *fk1[k]*cdt_2*curl_ez[i][j-1][k])
                    +cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fi2[i]*fj2[j-1]*(1+fk1[k])*iTA4[i][j-1][k]
                    -cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fi2[i]*(fj2[j-1]+fj2[j])*(1+fk1[k])*iTA4[i][j][k]
                    +cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fi2[i]*fj2[j]*(1+fk1[k])*iTA4[i][j+1][k];
            }

            TAf0]=0.;
            TAf[JE-1]=0.;
            TAe0]=0.;
            TAe[JE-1]=0.;

            for ( j=1; j < JE-1; j++ ) {
                TAf[j]=(TA4[j]+TA1[j]*TAf[j-1])/(TA2[j]-TA1[j]*TAe[j-1]);
                TAe[j]=TA3[j]/(TA2[j]-TA1[j]*TAe[j-1]);
            }

            dx[i][0][k]=0;
            dx[i][JE-1][k]=0;
            for ( j=JE-2; j>0; j--){
                dx[i][j][k]=TAf[j]+TAe[j]*dx[i][j+1][k];
            }
        }
    }

    /* Calculate the Dy field */
    for ( i=1; i < IE-1; i++ ) {
        for ( j=1; j < JE-2; j++ ) {
            for ( k=1; k < KE-1; k++ ) {
                B1=gay1[i][j][k]*sy1[i][j][k][NPR1]+gby1[i][j][k]*sy1[i][j][k][NPR2];
                B2=gay2[i][j][k]*sy2[i][j][k][NPR1]+gby2[i][j][k]*sy2[i][j][k][NPR2];
                B3=gay3[i][j][k]*sy3[i][j][k][NPR1]+gby3[i][j][k]*sy3[i][j][k][NPR2];
                B4=gay4[i][j][k]*sy4[i][j][k][NPR1]+gby4[i][j][k]*sy4[i][j][k][NPR2];
                iTA4[i][j][k] = gcy[i][j][k]*(- iy[i][j][k]-B1-B2-B3-B4);
            }
        }
    }

```

```

for ( i=1; i < IE-1; i++ ) {
for ( j=1; j < JE-2; j++ ) {
for ( k=1; k < KE-1; k++ ) {
TA1[k]=cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fj2[j]*fk2[k-1]*(1+fi1[i])*gcy[i][j][k-1];
TA2[k]=1+cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fj2[j]*(fk2[k-1]+fk2[k])*(1+fi1[i])*gcy[i][j][k];
TA3[k]=cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fj2[j]*fk2[k]*(1+fi1[i])*gcy[i][j][k+1];
curl_h = ( hx[i][j][k] - hx[i][j][k-1] - hzz[i][j][k] + hzz[i-1][j][k])/ddx ;
curl_hy[i][j][k]=curl_hy[i][j][k]+curl_h;
curl_e = ( ey[i][j][k+1] - ey[i][j][k] - ez[i][j+1][k] + ez[i][j][k])/ddx ;
curl_ex[i][j][k]=curl_ex[i][j][k]+curl_e;
TA4[k]=gk3[k]*gi3[i]*dy[i][j][k]-cdt_2dx*gk2[k]*gi2[i]*(hzz[i][j][k]-hzz[i-1][j][k])
+gk2[k]*gi2[i]*gj1[j]*cdt_2*curl_hy[i][j][k]+cdt_2dx*gk2[k]*gi2[i]*(1+gj1[j])
*(fj3[j]*fk3[k]*hx[i][j][k]-cdt_2dx*fj2[j]*fk2[k]*(ez[i][j+1][k]-ez[i][j][k])
+fj2[j]*fk2[k]*fi1[i]*cdt_2*curl_ex[i][j][k]-fj3[j]*fk3[k-1]*hx[i][j][k-1]
+cdt_2dx*fj2[j]*fk2[k-1]*(ez[i][j+1][k-1]-ez[i][j][k-1])-fj2[j]*fk2[k-1]
*fi1[i]*cdt_2*curl_ex[i][j][k-1])
+cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fj2[j]*fk2[k-1]*(1+fi1[i])*iTA4[i][j][k-1]
-cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fj2[j]*(fk2[k-1]+fk2[k])*(1+fi1[i])*iTA4[i][j][k]
+cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fj2[j]*fk2[k]*(1+fi1[i])*iTA4[i][j][k+1];
}

TAf[0]=0.;
TAf[KE-1]=0.;
TAe[0]=0.;
TAe[KE-1]=0.;

if (i>=ia&&i<=ib&&j>=ja&&j<=(jb-1)){
    TA4[ka]=TA4[ka]-cdt_2dx*hx_inc[j];
    TA4[kb]=TA4[kb]+cdt_2dx*hx_inc[j];
}

for ( k=1; k < KE-1; k++ ) {
    TAf[k]=(TA4[k]+TA1[k]*TAf[k-1])/(TA2[k]-TA1[k]*TAe[k-1]);
    TAc[k]=TA3[k]/(TA2[k]-TA1[k]*TAe[k-1]);
}

dy[i][j][0]=0.;
dy[i][j][KE-1]=0.;

for (k=KE-2; k>0; k--){
    dy[i][j][k]=TAf[k]+TAe[k]*dy[i][j][k+1];
}}

/* Calculate the Dz field */

for ( i=1; i < IE-1; i++ ) {
    for ( j=1; j < JE-1; j++ ) {
        for ( k=1; k < KE-2; k++ ) {
            B1=gaz1[i][j][k]*sz1[i][j][k][NPR1]+gbz1[i][j][k]*sz1[i][j][k][NPR2];
            B2=gaz2[i][j][k]*sz2[i][j][k][NPR1]+gbz2[i][j][k]*sz2[i][j][k][NPR2];
            B3=gaz3[i][j][k]*sz3[i][j][k][NPR1]+gbz3[i][j][k]*sz3[i][j][k][NPR2];
            B4=gaz4[i][j][k]*sz4[i][j][k][NPR1]+gbz4[i][j][k]*sz4[i][j][k][NPR2];
            iTA4[i][j][k] = gcz[i][j][k]*(- iz[i][j][k]-B1-B2-B3-B4);
        }}
}

```

```

for (j=1; j < JE-1; j++) {
for (k=1; k < KE-2; k++) {
for (i=1; i < IE-1; i++) {
TA1[i]=cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fk2[k]*fi2[i-1]*(1+fj1[j])*gcz[i-1][j][k];
TA2[i]=1+cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fk2[k]*(fi2[i-1]+fi2[i])*(1+fj1[j])*gcz[i][j][k];
TA3[i]=cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fk2[k]*fi2[i]*(1+fj1[j])*gcz[i+1][j][k];
curl_h = (hy[i][j][k] - hy[i-1][j][k] - hx[i][j][k] + hx[i-1][j][k])/ddx ;
curl_hz[i][j][k]=curl_hz[i][j][k]+curl_h;
curl_e = ( ezf[i+1][j][k] - ezf[i][j][k] - ex[i][j][k+1] + ex[i][j][k])/ddx ;
curl_ey[i][j][k]=curl_ey[i][j][k]+curl_e;
TA4[i]=g3[i]*gj3[j]*dz[i][j][k]-cdt_2dx*gi2[j]*gj2[j]*(hx[i][j][k]-hx[i][j-1][k])
+g2[i]*gj2[j]*gk1[k]*cdt_2*curl_hz[i][j][k]-cdt_2dx*gi2[j]*gj2[j]*(hx[i][j][k]-hx[i][j-1][k])
*(fk3[k]*fi3[i]*hy[i][j][k]-cdt_2dx*fk2[k]*fi2[i]*(ex[i][j][k+1]-ex[i][j][k])
+fk2[k]*fi2[i]*fj1[j]*cdt_2*curl_ey[i][j][k]-fk3[k]*fi3[i-1]*hy[i-1][j][k]
+cdt_2dx*fk2[k]*fi2[i-1]*(ex[i-1][j][k+1]-ex[i-1][j][k])-fk2[k]*fi2[i-1]
*fj1[j]*cdt_2*curl_ey[i-1][j][k])
+cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fk2[k]*fi2[i-1]*(1+fj1[j])*iTA4[i-1][j][k]
-cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fk2[k]*(fi2[i-1]+fi2[i])*(1+fj1[j])*iTA4[i][j][k]
+cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fk2[k]*fi2[i]*(1+fj1[j])*iTA4[i+1][j][k];
}
}
TAf[0]=0.;
TAf[IE-1]=0.;
TAe[0]=0.;
TAe[IE-1]=0.;
for (i=ia; i<=ib;i++){
if(k>=ka&&k<=kb-1&&j==ja)
{TA4[i]=TA4[i]+cdt_2dx*hx_inc_temp[ja-1];}
if (k>=ka&&k<=kb-1&&j==jb)
{TA4[i]=TA4[i]-cdt_2dx*hx_inc_temp[jb];}
}
if(j>=ja&&j<=jb&&k>=ka&&k<=kb-1){
TA4[ia-1]=TA4[ia-1]-cdt_2dx_sq*ez_inc[j];
TA4[ja]=TA4[ja]+cdt_2dx_sq*ez_inc[j];
TA4[jb]=TA4[jb]+cdt_2dx_sq*ez_inc[j];
TA4[ib+1]=TA4[ib+1]-cdt_2dx_sq*ez_inc[j];
}
}
for ( i=1; i < IE-1; i++) {
TAf[i]=(TA4[i]+TA1[i]*TAf[i-1])/(TA2[i]-TA1[i]*TAe[i-1]);
TAe[i]=(TA3[i]/(TA2[i]-TA1[i])*TAe[i-1]);
}
dz[0][j][k]=0.;
dz[IE-1][j][k]=0.;
for (i=IE-2; i>0; i--){
dz[i][j][k]=TAf[i]+TAe[i]*dz[i+1][j][k];
}}
/* Calculate the E from D field */
for ( i=1; i < IE-2; i++) {
for ( j=1; j < JE-1; j++) {
for ( k=1; k < KE-1; k++) {

```

```

ex_temp[i][j][k]=ex[i][j][k];/*for the calculation of h*/

B1=gax1[i][j][k]*sx1[i][j][k][NPR1]+gbx1[i][j][k]*sx1[i][j][k][NPR2];
B2=gax2[i][j][k]*sx2[i][j][k][NPR1]+gbx2[i][j][k]*sx2[i][j][k][NPR2];
B3=gax3[i][j][k]*sx3[i][j][k][NPR1]+gbx3[i][j][k]*sx3[i][j][k][NPR2];
B4=gax4[i][j][k]*sx4[i][j][k][NPR1]+gbx4[i][j][k]*sx4[i][j][k][NPR2];
ex[i][j][k] = gcx[i][j][k]*(dx[i][j][k] - ix[i][j][k]-B1-B2-B3-B4);
ix[i][j][k] = ix[i][j][k] + gx[i][j][k]*ex[i][j][k];
sx1[i][j][k][NCUR]=B1+gdx1[i][j][k]*ex[i][j][k];
sx2[i][j][k][NCUR]=B2+gdx2[i][j][k]*ex[i][j][k];
sx3[i][j][k][NCUR]=B3+gdx3[i][j][k]*ex[i][j][k];
sx4[i][j][k][NCUR]=B4+gdx4[i][j][k]*ex[i][j][k];
}}}

for ( i=1; i < IE-1; i++) {
    for ( j=1; j < JE-2; j++) {
        for ( k=1; k < KE-1; k++) {
            ey_temp[i][j][k]=ey[i][j][k];/*for the calculation of h*/

            B1=gay1[i][j][k]*sy1[i][j][k][NPR1]+gby1[i][j][k]*sy1[i][j][k][NPR2];
            B2=gay2[i][j][k]*sy2[i][j][k][NPR1]+gby2[i][j][k]*sy2[i][j][k][NPR2];
            B3=gay3[i][j][k]*sy3[i][j][k][NPR1]+gby3[i][j][k]*sy3[i][j][k][NPR2];
            B4=gay4[i][j][k]*sy4[i][j][k][NPR1]+gby4[i][j][k]*sy4[i][j][k][NPR2];
            ey[i][j][k] = gcy[i][j][k]*(dy[i][j][k] - iy[i][j][k]-B1-B2-B3-B4);
            iy[i][j][k] = iy[i][j][k] + gy[i][j][k]*ey[i][j][k];
            sy1[i][j][k][NCUR]=B1+gdy1[i][j][k]*ey[i][j][k];
            sy2[i][j][k][NCUR]=B2+gdy2[i][j][k]*ey[i][j][k];
            sy3[i][j][k][NCUR]=B3+gdy3[i][j][k]*ey[i][j][k];
            sy4[i][j][k][NCUR]=B4+gdy4[i][j][k]*ey[i][j][k];
        }}}

    for ( i=1; i < IE-1; i++) {
        for ( j=1; j < JE-1; j++) {
            for ( k=1; k < KE-2; k++) {
                ez_temp[i][j][k]=ez[i][j][k];/*for the calculation of h*/

                B1=gaz1[i][j][k]*sz1[i][j][k][NPR1]+gbz1[i][j][k]*sz1[i][j][k][NPR2];
                B2=gaz2[i][j][k]*sz2[i][j][k][NPR1]+gbz2[i][j][k]*sz2[i][j][k][NPR2];
                B3=gaz3[i][j][k]*sz3[i][j][k][NPR1]+gbz3[i][j][k]*sz3[i][j][k][NPR2];
                B4=gaz4[i][j][k]*sz4[i][j][k][NPR1]+gbz4[i][j][k]*sz4[i][j][k][NPR2];
                ez[i][j][k] = gcx[i][j][k]*(dz[i][j][k] - iz[i][j][k]-B1-B2-B3-B4);
                iz[i][j][k] = iz[i][j][k] + gz[i][j][k]*ez[i][j][k];
                sz1[i][j][k][NCUR]=B1+gdz1[i][j][k]*ez[i][j][k];
                sz2[i][j][k][NCUR]=B2+gdz2[i][j][k]*ez[i][j][k];
                sz3[i][j][k][NCUR]=B3+gdz3[i][j][k]*ez[i][j][k];
                sz4[i][j][k][NCUR]=B4+gdz4[i][j][k]*ez[i][j][k];
            }}}

/* Calculate the Hx field */
for ( i=0; i < ia; i++) {
    for ( j=0; j < JE-1; j++) {
        for ( k=0; k < KE-1; k++) {
            hx[i][j][k] = fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(1+f1[i])*(ey[i][j][k+1]-ey[i][j][k])
                -cdt_2dx*fj2[j]*fk2[k]*(ez_temp[i][j+1][k]-ez_temp[i][j][k])
                +fj2[j]*fk2[k]*f1[i]*cdt_2*curl_ex[i][j][k];
        } } }

```

```

for ( i=ia; i <= ib; i++ ) {
for ( j=0; j < JE-1; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hx[i][j][k] = fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(1+fi1[i])*(ey[i][j][k+1]-ey[i][j][k])
-cdt_2dx*fj2[j]*fk2[k]*(ez_temp[i][j+1][k]-ez_temp[i][j][k]);
} } }

for ( i=ib+1; i < IE; i++ ) {
ixh = i - ib-1;
for ( j=0; j < JE-1; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hx[i][j][k] = fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(1+fi1[i])*(ey[i][j][k+1]-ey[i][j][k])
-cdt_2dx*fj2[j]*fk2[k]*(ez_temp[i][j+1][k]-ez_temp[i][j][k])
+fj2[j]*fk2[k]*fi1[i]*cdt_2*curl_ex[i][j][k];
} } }

/* Incident Hx */
for ( i=ia; i <= ib; i++ ) {
for ( k=ka; k <= kb-1; k++ ) {
hx[i][ja-1][k] = hx[i][ja-1][k] + cdt_2dx*ez_inc_temp[ja];
hx[i][jb][k] = hx[i][jb][k] - cdt_2dx*ez_inc_temp[jb];
} }

/* Calculate the Hy field */
for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < ja; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hy[i][j][k] = fk3[k]*fi3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*fi2[i]*(1+fj1[j])*(ez[i+1][j][k]-ez[i][j][k])
-cdt_2dx*fk2[k]*fi2[i]*(ex_temp[i][j][k+1]-ex_temp[i][j][k])
+fk2[k]*fi2[i]*fj1[j]*cdt_2*curl_ey[i][j][k];
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=ja; j <= jb; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hy[i][j][k] = fk3[k]*fi3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*fi2[i]*(1+fj1[j])*(ez[i+1][j][k]-ez[i][j][k])
-cdt_2dx*fk2[k]*fi2[i]*(ex_temp[i][j][k+1]-ex_temp[i][j][k]);
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=jb+1; j < JE; j++ ) {
jyh = j - jb-1;
for ( k=0; k < KE-1; k++ ) {
hy[i][j][k] = fk3[k]*fi3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*fi2[i]*(1+fj1[j])*(ez[i+1][j][k]-ez[i][j][k])
-cdt_2dx*fk2[k]*fi2[i]*(ex_temp[i][j][k+1]-ex_temp[i][j][k])
+fk2[k]*fi2[i]*fj1[j]*cdt_2*curl_ey[i][j][k];
} } }

/* Incident Hy */
for ( j=ja; j <= jb; j++ ) {
for ( k=ka; k <= kb-1; k++ ) {
hy[ia-1][j][k] = hy[ia-1][j][k] -cdt_2dx*ez_inc[j];
hy[ib][j][k] = hy[ib][j][k] + cdt_2dx*ez_inc[j];
} }

```

```

/* Calculate the Hz field */
for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < JE-1; j++ ) {
for ( k=0; k < ka; k++ ) {
hzz[i][j][k] = fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(1+fk1[k])*(ex[i][j+1][k]-ex[i][j][k])
-cdt_2dx*fi2[i]*fj2[j]*(ey_temp[i+1][j][k]-ey_temp[i][j][k])
+fi2[i]*fj2[j]*fk1[k]*cdt_2*curl_ey[i][j][k];
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < JE-1; j++ ) {
for ( k=ka; k <= kb; k++ ) {
hzz[i][j][k] = fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(1+fk1[k])*(ex[i][j+1][k]-ex[i][j][k])
-cdt_2dx*fi2[i]*fj2[j]*(ey_temp[i+1][j][k]-ey_temp[i][j][k]);
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < JE-1; j++ ) {
for ( k=kb+1; k < KE; k++ ) {
kzh = k - kb - 1;
hzz[i][j][k] = fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(1+fk1[k])*(ex[i][j+1][k]-ex[i][j][k])
-cdt_2dx*fi2[i]*fj2[j]*(ey_temp[i+1][j][k]-ey_temp[i][j][k])
+fi2[i]*fj2[j]*fk1[k]*cdt_2*curl_ey[i][j][k];
} } }

//n=n+0.5;
T = T + 0.5;
NPR2=NPR1;
NPR1=NCUR;
NCUR=(NCUR+1)%3;

/* Calculate the incident buffer */

for ( j=1; j < JE-1; j++ ) {
B1=inc_gaz1[j]*inc_sz1[j][NPR1]+inc_gbz1[j]*inc_sz1[j][NPR2];
B2=inc_gaz2[j]*inc_sz2[j][NPR1]+inc_gbz2[j]*inc_sz2[j][NPR2];
B3=inc_gaz3[j]*inc_sz3[j][NPR1]+inc_gbz3[j]*inc_sz3[j][NPR2];
B4=inc_gaz4[j]*inc_sz4[j][NPR1]+inc_gbz4[j]*inc_sz4[j][NPR2];
inc_iTA4[j] = inc_gcz[j]*(- iz_inc[j]-B1-B2-B3-B4);
}

for ( j=1; j < JE-1; j++ ) {
TA1[j]=cdt_2dx_sq*inc_gcz[j-1];
TA2[j]=1+cdt_2dx_sq*2.0*inc_gcz[j];
TA3[j]=cdt_2dx_sq*inc_gcz[j+1];
TA4[j]=dz_inc[j]+cdt_2dx*( hx_inc[j-1]- hx_inc[j])
+cdt_2dx_sq*inc_iTA4[j-1]
-cdt_2dx_sq*2.0*inc_iTA4[j]
+cdt_2dx_sq*inc_iTA4[j+1];
}

TAf[0]=0.;
TAf[JE-1]=0.;
TAef[0]=0.;
TAef[JE-1]=0.;

```

```

for (j=1; j < JE-1; j++) {
    TAf[j]=(TA4[j]+TA1[j]*TAf[j-1])/(TA2[j]-TA1[j]*TAe[j-1]);
    TAe[j]=TA3[j]/(TA2[j]-TA1[j]*TAe[j-1]);
}

dz_inc[0]=0.;
dz_inc[JE-1]=0.;
for (j=JE-2; j>0; j--){
    dz_inc[j]=TAf[j]+TAe[j]*dz_inc[j+1];
}

pulse=-1.0e6*(100.-T)/sqrt(200.)*exp(-(T-100.)*(T-100.)/400.0);
dz_inc[3]=pulse;

for (j=0; j < JE; j++) {
    ez_inc_temp[j]=ez_inc[j];
}

for (j=1; j < JE-1; j++) {
    B1=inc_gaz1[j]*inc_sz1[j][NPR1]+inc_gbz1[j]*inc_sz1[j][NPR2];
    B2=inc_gaz2[j]*inc_sz2[j][NPR1]+inc_gbz2[j]*inc_sz2[j][NPR2];
    B3=inc_gaz3[j]*inc_sz3[j][NPR1]+inc_gbz3[j]*inc_sz3[j][NPR2];
    B4=inc_gaz4[j]*inc_sz4[j][NPR1]+inc_gbz4[j]*inc_sz4[j][NPR2];
    ez_inc[j] = inc_gcz[j]*(dz_inc[j] - iz_inc[j]-B1-B2-B3-B4);
    iz_inc[j]= iz_inc[j] + inc_gz[j]*ez_inc[j];
    inc_sz1[j][NCUR]=B1+inc_gdz1[j]*ez_inc[j];
    inc_sz2[j][NCUR]=B2+inc_gdz2[j]*ez_inc[j];
    inc_sz3[j][NCUR]=B3+inc_gdz3[j]*ez_inc[j];
    inc_sz4[j][NCUR]=B4+inc_gdz4[j]*ez_inc[j];
}

/* Boundary conditions for the incident buffer*/
ez_inc[0] = ez_low_m2;
ez_low_m2 = ez_low_m1;
ez_low_m1 = ez_inc[1];

ez_inc[JE-1] = ez_high_m2;
ez_high_m2 = ez_high_m1;
ez_high_m1 = ez_inc[JE-2];

for (j=0; j < JE; j++) {
    hx_inc_temp[j]=hx_inc[j];
}

for (j=0; j < JE-1; j++) {
    hx_inc[j] = hx_inc[j] + cdt_2dx*( ez_inc[j] - ez_inc[j+1] );
}

/* Calculate the Dx field */
for (i=1; i < IE-2; i++) {
    for (j=1; j < JE-1; j++) {
        for (k=1; k < KE-1; k++) {
            B1=gax1[i][j][k]*sx1[i][j][k][NPR1]+gbx1[i][j][k]*sx1[i][j][k][NPR2];
            B2=gax2[i][j][k]*sx2[i][j][k][NPR1]+gbx2[i][j][k]*sx2[i][j][k][NPR2];
            B3=gax3[i][j][k]*sx3[i][j][k][NPR1]+gbx3[i][j][k]*sx3[i][j][k][NPR2];
            B4=gax4[i][j][k]*sx4[i][j][k][NPR1]+gbx4[i][j][k]*sx4[i][j][k][NPR2];

```

```

iTA4[i][j][k] = gcx[i][j][k]*(-ix[i][j][k]-B1-B2-B3-B4);
}}

for ( i=1; i < IE-2; i++ ) {
for ( j=1; j < JE-1; j++ ) {
for ( k=1; k < KE-1; k++ ) {
TA1[k]=cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*f2[k-1]*f2[i]*(1+fj1[j])*gcx[i][j][k-1];
TA2[k]=1+cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*(fk2[k]+fk2[k-1])*f2[i]*(1+fj1[j])*gcx[i][j][k];
TA3[k]=cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fk2[k]*f2[i]*(1+fj1[j])*gcx[i][j][k+1];
curl_h = ( hzz[i][j][k] - hzz[i][j]-1][k] - hy[i][j][k] + hy[i][j][k-1])/ddx ;
curl_hx[i][j][k]=curl_hx[i][j][k]+curl_h;
curl_e = ( ez[i+1][j][k] - ez[i][j][k] - ex[i][j][k+1] + ex[i][j][k])/ddx ;
curl_ey[i][j][k]=curl_ey[i][j][k]+curl_e;
TA4[k]=gj3[j]*gk3[k]*dx[i][j][k]+cdt_2dx*gj2[j]*gk2[k]*(hzz[i][j][k]-hzz[i][j]-1][k]
+gj2[j]*gk2[k]*gi1[i])*cdt_2*curl_hx[i][j][k]-cdt_2dx*gj2[j]*gk2[k]*(1+gi1[i])
*(fk3[k]*fj3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*f2[i]*(ez[i+1][j][k]-ez[i][j][k])
+fk2[k]*f2[i]*fj1[j])*cdt_2*curl_ey[i][j][k]-fk3[k-1]*f3[i]*hy[i][j][k-1]
-cdt_2dx*fk2[k-1]*f2[i]*(ez[i+1][j][k-1]-ez[i][j][k-1])-fk2[k-1]
*f2[i]*fj1[j])*cdt_2*curl_ey[i][j][k-1])
+cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fk2[k-1]*f2[i]*(1+fj1[j])*TA4f[i][j][k-1]
-cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*(fk2[k]+fk2[k-1])*f2[i]*(1+fj1[j])*TA4f[i][j][k]
+cdt_2dx_sq*gj2[j]*gk2[k]*(1+gi1[i])*fk2[k]*f2[i]*(1+fj1[j])*TA4f[i][j][k+1];
}

if(i>=ja&&j<=jb){
if(i==ia-1){TA4[ka]=TA4[kb]+ez_inc_temp[j];
TA4[kb]=TA4[kb]-ez_inc_temp[j];
}
if(i==ib){TA4[ka]=TA4[kb]-ez_inc_temp[j];
TA4[kb]=TA4[kb]+ez_inc_temp[j];
}
}

TAf[0]=0.;
TAf[KE-1]=0.;
TAe[0]=0.;
TAe[KE-1]=0.;

for ( k=1; k < KE-1; k++ ) {
TAf[k]=(TA4[k]+TA1[k]*TAf[k-1])/(TA2[k]-TA1[k]*TAe[k-1]);
TAe[k]=TA3[k]/(TA2[k]-TA1[k]*TAe[k-1]);
}

dx[i][j][0]=0.;
dx[i][j][KE-1]=0.;
for ( k=KE-2; k>0; k--){
dx[i][j][k]=TAf[k]+TAe[k]*dx[i][j][k+1];
}}

/* Calculate the Dy field */
for ( i=1; i < IE-1; i++ ) {
for ( j=1; j < JE-2; j++ ) {
for ( k=1; k < KE-1; k++ ) {
B1=gay1[i][j][k]*sy1[i][j][k][NPR1]+gby1[i][j][k]*sy1[i][j][k][NPR2];
B2=gay2[i][j][k]*sy2[i][j][k][NPR1]+gby2[i][j][k]*sy2[i][j][k][NPR2];
B3=gay3[i][j][k]*sy3[i][j][k][NPR1]+gby3[i][j][k]*sy3[i][j][k][NPR2];
}
}
}

```

```

B4=gay4[i][j][k]*sy4[i][j][k][NPR1]+gby4[i][j][k]*sy4[i][j][k][NPR2];
iTA4[i][j][k] = gcy[i][j][k]*(- iy[i][j][k]-B1-B2-B3-B4);
}}}

for ( j=1; j < JE-2; j++ ) {
for ( k=1; k < KE-1; k++ ) {
for ( i=1; i < IE-1; i++ ) {
TA1[i]=cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fi2[i-1]*fj2[j]*(1+fk1[k])*gcy[i-1][j][k];
TA2[i]=1+cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*(fi2[i]+fi2[i-1])*fj2[j]*(1+fk1[k])*gcy[i][j][k];
TA3[i]=cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fi2[i]*fj2[j]*(1+fk1[k])*gcy[i+1][j][k];
curl_h = ( hx[i][j][k] - hx[i][j][k-1]- hzz[i][j][k] + hzz[i-1][j][k])/ddx ;
curl_hy[i][j][k]=curl_hy[i][j][k]+curl_h;
curl_e=( ex[i][j+1][k] - ex[i][j][k]- ey[i+1][j][k] + ey[i][j][k])/ddx ;
curl_ez[i][j][k]=curl_ez[i][j][k]+curl_e;
TA4[i]=gk3[k]*gi3[i]*dy[i][j][k]+cdt_2dx*gk2[k]*gi2[i]*(hx[i][j][k]-hx[i][j][k-1])
+gk2[k]*gi2[i]*gj1[j]*cdt_2*curl_hy[i][j][k]-cdt_2dx*gk2[k]*gi2[i]*(1+gj1[j])
*(fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(ex[i][j+1][k]-ex[i][j][k])
+fi2[i]*fj2[j]*fk1[k]*cdt_2*curl_ez[i][j][k]-fi3[i-1]*fj3[j]*hzz[i-1][j][k]
-cdt_2dx*fi2[i-1]*fj2[j]*(ex[i-1][j+1][k]-ex[i-1][j][k])-fi2[i-1]
*fj2[j]*fk1[k]*cdt_2*curl_ez[i-1][j][k])
+cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fi2[i-1]*fj2[j]*(1+fk1[k])*iTA4[i-1][j][k]
-cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*(fi2[i]+fi2[i-1])*fj2[j]*(1+fk1[k])*iTA4[i][j][k]
+cdt_2dx_sq*gk2[k]*gi2[i]*(1+gj1[j])*fi2[i]*fj2[j]*(1+fk1[k])*iTA4[i+1][j][k];
}
}

for(i=ia;i<=ib;i++){
if (j>=ja&&j<=(jb-1)&&k==ka){
TA4[i]=TA4[i]-cdt_2dx*hx_inc_temp[j];}
if (j>=ja&&j<=(jb-1)&&k==kb){
TA4[i]=TA4[i]+cdt_2dx*hx_inc_temp[j];
}
}

TAf[0]=0.;
TAf[IE-1]=0.;
TAe[0]=0.;
TAe[IE-1]=0.;

for ( i=1; i < IE-1; i++ ) {
TAf[i]=(TA4[i]+TA1[i]*TAf[i-1])/(TA2[i]-TA1[i]*TAe[i-1]);
TAe[i]=TA3[i]/(TA2[i]-TA1[i]*TAe[i-1]);
}

dy[0][j][k]=0.;
dy[IE-1][j][k]=0.;
for (i=IE-2; i>0; i--){
dy[i][j][k]=TAf[i]+TAe[i]*dy[i+1][j][k];
}}}

/* Calculate the Dz field */

for ( i=1; i < IE-1; i++ ) {
for ( j=1; j < JE-1; j++ ) {
for ( k=1; k < KE-2; k++ ) {
B1=gaz1[i][j][k]*sz1[i][j][k][NPR1]+gbz1[i][j][k]*sz1[i][j][k][NPR2];
B2=gaz2[i][j][k]*sz2[i][j][k][NPR1]+gbz2[i][j][k]*sz2[i][j][k][NPR2];

```

```

B3=gaz3[i][j][k]*sz3[i][j][k][NPR1]+gbz3[i][j][k]*sz3[i][j][k][NPR2];
B4=gaz4[i][j][k]*sz4[i][j][k][NPR1]+gbz4[i][j][k]*sz4[i][j][k][NPR2];
iTA4[i][j][k] = gcz[i][j][k]*(- iz[i][j][k]-B1-B2-B3-B4);
}}}

for ( i=1; i < IE-1; i++) {
for (k=1; k < KE-2; k++ ) {
for (j=1; j < JE-1; j++ ) {
TA1[j]=cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fj2[j-1]*fk2[k]*(1+fi1[i])*gcz[i][j-1][k];
TA2[j]=1+cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*(fj2[j]+fj2[j-1])*fk2[k]*(1+fi1[i])*gcz[i][j][k];
TA3[j]=cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fj2[j]*fk2[k]*(1+fi1[i])*gcz[i][j+1][k];
curl_h = ( hy[i][j][k] - hy[i-1][j][k]- hx[i][j][k] + hx[i][j-1][k])/ddx ;
curl_hz[i][j][k]=curl_hz[i][j][k]+curl_h;
curl_e= ( ey[i][j][k+1] - ey[i][j][k]- ez[i][j+1][k] + ez[i][j][k])/ddx ;
curl_ex[i][j][k]=curl_ex[i][j][k]+curl_e;
TA4[j]=gi3[i]*gj3[j]*dz[i][j][k]+cdt_2dx*gi2[i]*gj2[j]*(hy[i][j][k]-hy[i-1][j][k])
+gi2[i]*gj2[j]*gk1[k]*cdt_2*curl_hz[i][j][k]-cdt_2dx*gi2[i]*gj2[j]*(1+gk1[k])
*(fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(ey[i][j][k+1]-ey[i][j][k])
+fj2[j]*fk2[k]*fi1[i]*cdt_2*curl_ex[i][j][k]-fj3[j-1]*fk3[k]*hx[i][j-1][k]
-cdt_2dx*fj2[j-1]*fk2[k]*(ey[i][j-1][k+1]-ey[i][j-1][k])-fj2[j-1]
*fk2[k]*fi1[i]*cdt_2*curl_ex[i][j-1][k])
+cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fj2[j-1]*fk2[k]*(1+fi1[i])*iTA4[i][j-1][k]
-cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*(fj2[j]+fj2[j-1])*fk2[k]*(1+fi1[i])*iTA4[i][j][k]
+cdt_2dx_sq*gi2[i]*gj2[j]*(1+gk1[k])*fj2[j]*fk2[k]*(1+fi1[i])*iTA4[i][j+1][k];
}

TAf[0]=0.;
TAf[JE-1]=0.;
TAe[0]=0.;
TAe[JE-1]=0.;

if(k>=ka&&k<=kb-1&&i>=ia&&i<=ib){
TA4[ja]=TA4[ja]+cdt_2dx*hx_inc[ja-1];
TA4[jb]=TA4[jb]-cdt_2dx*hx_inc[jb];}

for ( j=1; j < JE-1; j++ ) {
TAf[j]=(TA4[j]+TA1[j]*TAf[j-1])/(TA2[j]-TA1[j]*TAe[j-1]);
TAe[j]=TA3[j]/(TA2[j]-TA1[j]*TAe[j-1]);
}

dz[i][0][k]=0.;
dz[i][JE-1][k]=0.;
for (j=JE-2; j>0; j--){
dz[i][j][k]=TAf[j]+TAe[j]*dz[i][j+1][k];
}}}

/* Calculate the E from D field */

for ( i=1; i < IE-2; i++ ) {
for ( j=1; j < JE-1; j++ ) {
for ( k=1; k < KE-1; k++ ) {
ex_temp[i][j][k]=ex[i][j][k];/*for the calculation of h*/

B1=gax1[i][j][k]*sx1[i][j][k][NPR1]+gbx1[i][j][k]*sx1[i][j][k][NPR2];
B2=gax2[i][j][k]*sx2[i][j][k][NPR1]+gbx2[i][j][k]*sx2[i][j][k][NPR2];

```

```

B3=gax3[i][j][k]*sx3[i][j][k][NPR1]+gbx3[i][j][k]*sx3[i][j][k][NPR2];
B4=gax4[i][j][k]*sx4[i][j][k][NPR1]+gbx4[i][j][k]*sx4[i][j][k][NPR2];
ex[i][j][k] = gcx[i][j][k]*(dx[i][j][k] - ix[i][j][k]-B1-B2-B3-B4);
ix[i][j][k] = ix[i][j][k] + gx[i][j][k]*ex[i][j][k];
sx1[i][j][k][NCUR]=B1+gdx1[i][j][k]*ex[i][j][k];
sx2[i][j][k][NCUR]=B2+gdx2[i][j][k]*ex[i][j][k];
sx3[i][j][k][NCUR]=B3+gdx3[i][j][k]*ex[i][j][k];
sx4[i][j][k][NCUR]=B4+gdx4[i][j][k]*ex[i][j][k];
}}}

for ( i=1; i < IE-1; i++ ) {
  for ( j=1; j < JE-2; j++ ) {
    for ( k=1; k < KE-1; k++ ) {
      ey_temp[i][j][k]=ey[i][j][k];/*for the calculation of h*/

      B1=gay1[i][j][k]*sy1[i][j][k][NPR1]+gby1[i][j][k]*sy1[i][j][k][NPR2];
      B2=gay2[i][j][k]*sy2[i][j][k][NPR1]+gby2[i][j][k]*sy2[i][j][k][NPR2];
      B3=gay3[i][j][k]*sy3[i][j][k][NPR1]+gby3[i][j][k]*sy3[i][j][k][NPR2];
      B4=gay4[i][j][k]*sy4[i][j][k][NPR1]+gby4[i][j][k]*sy4[i][j][k][NPR2];
      ey[i][j][k] = gcy[i][j][k]*(dy[i][j][k] - iy[i][j][k]-B1-B2-B3-B4);
      iy[i][j][k] = iy[i][j][k] + gy[i][j][k]*ey[i][j][k];
      sy1[i][j][k][NCUR]=B1+gdy1[i][j][k]*ey[i][j][k];
      sy2[i][j][k][NCUR]=B2+gdy2[i][j][k]*ey[i][j][k];
      sy3[i][j][k][NCUR]=B3+gdy3[i][j][k]*ey[i][j][k];
      sy4[i][j][k][NCUR]=B4+gdy4[i][j][k]*ey[i][j][k];
    }}}

for ( i=1; i < IE-1; i++ ) {
  for ( j=1; j < JE-1; j++ ) {
    for ( k=1; k < KE-2; k++ ) {
      ez_temp[i][j][k]=ez[i][j][k];/*for the calculation of h*/

      B1=gaz1[i][j][k]*sz1[i][j][k][NPR1]+gbz1[i][j][k]*sz1[i][j][k][NPR2];
      B2=gaz2[i][j][k]*sz2[i][j][k][NPR1]+gbz2[i][j][k]*sz2[i][j][k][NPR2];
      B3=gaz3[i][j][k]*sz3[i][j][k][NPR1]+gbz3[i][j][k]*sz3[i][j][k][NPR2];
      B4=gaz4[i][j][k]*sz4[i][j][k][NPR1]+gbz4[i][j][k]*sz4[i][j][k][NPR2];
      ez[i][j][k] = gcz[i][j][k]*(dz[i][j][k] - iz[i][j][k]-B1-B2-B3-B4);
      iz[i][j][k] = iz[i][j][k] + gz[i][j][k]*ez[i][j][k];
      sz1[i][j][k][NCUR]=B1+gdz1[i][j][k]*ez[i][j][k];
      sz2[i][j][k][NCUR]=B2+gdz2[i][j][k]*ez[i][j][k];
      sz3[i][j][k][NCUR]=B3+gdz3[i][j][k]*ez[i][j][k];
      sz4[i][j][k][NCUR]=B4+gdz4[i][j][k]*ez[i][j][k];
    }}}

/* Calculate the Hx field */
for ( i=0; i < ia; i++ ) {
  for ( j=0; j < JE-1; j++ ) {
    for ( k=0; k < KE-1; k++ ) {
      hx[i][j][k] = fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(ey_temp[i][j][k+1]-ey_temp[i][j][k])
        -cdt_2dx*fj2[j]*fk2[k]*(1+fi1[i])* (ez[i][j+1][k]-ez[i][j][k])
        +fj2[j]*fk2[k]*fi1[i]*cdt_2*curl_ex[i][j][k];
    } } }

for ( i=ia; i <= ib; i++ ) {
  for ( j=0; j < JE-1; j++ ) {
    for ( k=0; k < KE-1; k++ ) {

```

```

hx[i][j][k] = fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(ey_temp[i][j][k+1]-ey_temp[i][j][k])
             -cdt_2dx*fj2[j]*fk2[k]*(1+fi1[i])*(ez[i][j+1][k]-ez[i][j][k]);
} } }

for ( i=ib+1; i < IE; i++ ) {
ixh = i - ib-1;
for ( j=0; j < JE-1; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hx[i][j][k] = fj3[j]*fk3[k]*hx[i][j][k]+cdt_2dx*fj2[j]*fk2[k]*(ey_temp[i][j][k+1]-ey_temp[i][j][k])
             -cdt_2dx*fj2[j]*fk2[k]*(1+fi1[i])*(ez[i][j+1][k]-ez[i][j][k])
             +fj2[j]*fk2[k]*fi1[i]*cdt_2*curl_ex[i][j][k];
} } }

/* Incident Hx */
for ( i=ia; i <= ib; i++ ) {
for ( k=ka; k <= kb-1; k++ ) {
hx[i][ja-1][k] = hx[i][ja-1][k] + cdt_2dx*ez_inc[ja];
hx[i][jb][k] = hx[i][jb][k] - cdt_2dx*ez_inc[jb];
} }

/* Calculate the Hy field */
for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < ja; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hy[i][j][k] = fk3[k]*fi3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*fi2[i]*(ez_temp[i+1][j][k]-ez_temp[i][j][k])
             -cdt_2dx*fk2[k]*fi2[i]*(1+fj1[j])*(ex[i][j][k+1]-ex[i][j][k])
             +fk2[k]*fi2[i]*fj1[j]*cdt_2*curl_ey[i][j][k];
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=ja; j <= jb; j++ ) {
for ( k=0; k < KE-1; k++ ) {
hy[i][j][k] = fk3[k]*fi3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*fi2[i]*(ez_temp[i+1][j][k]-ez_temp[i][j][k])
             -cdt_2dx*fk2[k]*fi2[i]*(1+fj1[j])*(ex[i][j][k+1]-ex[i][j][k]);
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=jb+1; j < JE; j++ ) {
jyh = j - jb-1;
for ( k=0; k < KE-1; k++ ) {
hy[i][j][k] = fk3[k]*fi3[i]*hy[i][j][k]+cdt_2dx*fk2[k]*fi2[i]*(ez_temp[i+1][j][k]-ez_temp[i][j][k])
             -cdt_2dx*fk2[k]*fi2[i]*(1+fj1[j])*(ex[i][j][k+1]-ex[i][j][k])
             +fk2[k]*fi2[i]*fj1[j]*cdt_2*curl_ey[i][j][k];
} } }

/* Incident Hy */
for ( j=ja; j <= jb; j++ ) {
for ( k=ka; k <= kb-1; k++ ) {
hy[ia-1][j][k] = hy[ia-1][j][k] - cdt_2dx*ez_inc[j];
hy[ib][j][k] = hy[ib][j][k] + cdt_2dx*ez_inc[j];
} }

/* Calculate the Hz field */
for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < JE-1; j++ ) {

```

```

for ( k=0; k < ka; k++ ) {
hzz[i][j][k] = fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(ex_temp[i][j+1][k]-ex_temp[i][j][k])
-cdt_2dx*fi2[i]*fj2[j]*(1+fk1[k])*(ey[i+1][j][k]-ey[i][j][k])
+fi2[i]*fj2[j]*fk1[k]*cdt_2*curl_ey[i][j][k];
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < JE-1; j++ ) {
for ( k=ka; k <= kb; k++ ) {
hzz[i][j][k] = fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(ex_temp[i][j+1][k]-ex_temp[i][j][k])
-cdt_2dx*fi2[i]*fj2[j]*(1+fk1[k])*(ey[i+1][j][k]-ey[i][j][k]);
} } }

for ( i=0; i < IE-1; i++ ) {
for ( j=0; j < JE-1; j++ ) {
for ( k=kb+1; k < KE; k++ ) {
hzz[i][j][k] = fi3[i]*fj3[j]*hzz[i][j][k]+cdt_2dx*fi2[i]*fj2[j]*(ex_temp[i][j+1][k]-ex_temp[i][j][k])
-cdt_2dx*fi2[i]*fj2[j]*(1+fk1[k])*(ey[i+1][j][k]-ey[i][j][k])
+fi2[i]*fj2[j]*fk1[k]*cdt_2*curl_ey[i][j][k];
} } }

if(n==(int)(120)){
    fprintf(fp4,"ZONE I=61, J=61, F=POINT\n");
    int ks=(int)((IE-1)/2);
    for(int ys=0;ys<JE;ys++)
    for(int xs=0;xs<IE;xs++)
        fprintf(fp4,"%d %d %lf\n",xs,ys,ez[xs][ys][ks]);
}

if(n==(int)(144)){
    fprintf(fp5,"ZONE I=61, J=61, F=POINT\n");
    int ks=(int)((IE-1)/2);
    for(int ys=0;ys<JE;ys++)
    for(int xs=0;xs<IE;xs++)
        fprintf(fp5,"%d %d %lf\n",xs,ys,ez[xs][ys][ks]);
}

if(n==(int)(168)){
    fprintf(fp6,"ZONE I=61, J=61, F=POINT\n");
    int ks=(int)((IE-1)/2);
    for(int ys=0;ys<JE;ys++)
    for(int xs=0;xs<IE;xs++)
        fprintf(fp6,"%d %d %lf\n",xs,ys,ez[xs][ys][ks]);
}

if(n==(int)(192)){
    fprintf(fp7,"ZONE I=61, J=61, F=POINT\n");
    int ks=(int)((IE-1)/2);
    for(int ys=0;ys<JE;ys++)
    for(int xs=0;xs<IE;xs++)
        fprintf(fp7,"%d %d %lf\n",xs,ys,ez[xs][ys][ks]);
}

if(n==(int)(216)){
    fprintf(fp8,"ZONE I=61, J=61, F=POINT\n");
}

```

```
        int ks=(int)((IE-1)/2);
        for(int ys=0;ys<JE;ys++)
        for(int xs=0;xs<IE;xs++)
            fprintf(fp8,"%d  %d  %lf\n",xs,ys,ez[xs][ys][ks]);
    }
}

fclose(fp4);fclose(fp5);fclose(fp6); fclose(fp7); fclose(fp8);

/* ---- End of the main FDTD loop ---- */

    return 0;
}
```

REFERENCES

- [1] Federal Communication Commission (FCC). "First report and order in the matter of revision of part 15 of the commission's rules regarding ultra-wideband transmission systems," ET Docket 98-153, FCC 02-48, adopted in Feb. 2002, released in April 2002.
- [2] R. Albanese, J. Blaschak, R. Medina, and J. Penn, "Ultrashort electromagnetic signals: biophysical questions, safety issues, and medical opportunities," *Aviat. Space Environ. Med.* vol. 65, pp. A116-120, 1994.
- [3] K. S. Kunz and R. J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*. Boca Raton, FL: CRC Press, 1993.
- [4] S. Samn and S. P. Mathur, "A mathematical model of a gigahertz transverse electromagnetic cell, I," *AFRL-HE-TR-1999-0219*, pp.1-18, 1999.
- [5] K. H. Schoenbach, S. J. Beebe, and E. S. Buescher, "Intracellular effect of ultrashort electrical pulses," *Bioelectromagnetics*, vol. 22, pp. 40-448, 2001.
- [6] R. P. Joshi, Q. Hu, K. H. Schoenbach, and H. P. Hjalmarson, "Improved energy model for membrane electroporation in biological cells subjected to electrical pulses," *Phys. Rev. E.*, vol. 65, pp. 041920:01-041920:07, 2002.
- [7] R. P. Joshi, Q. Hu, K. H. Schoenbach, and S. J. Beebe, "Simulations of electroporation dynamics and shape deformations in biological cells subjected to high voltage pulses," *IEEE Transactions on Plasma Science*, vol. 30, pp. 1536-1546, 2002.
- [8] K. S. Yee, "Numerical solution of initial boundary value problem involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas and Propagation*, vol. 14, pp. 302-307, 1966.
- [9] A. Taflove, *Computational Electrodynamics: The Finite Difference Time Domain Method*. Boston, MA: Artech House, 1995.
- [10] J. Lee and B. Fornberg, "A split step approach for the 3-D Maxwell's equations," *J. Comput. Appl. Math.*, vol. 158, pp. 485-505, 2003.

- [11] W. Fu and E. L. Tan, "Development of split-step FDTD method with higher order spatial accuracy," *Electron. Lett.*, vol. 40, pp. 1252-1254, 2004.
- [12] J. Shibayama, M. Muraki, J. Yamauchi, and H. Nakano, "Efficient implicit FDTD algorithm based on locally one-dimensional scheme," *Electron. Lett.*, vol. 41, pp. 1046-1047, 2005.
- [13] V. Ed. Nascimento, B. H. V. Borges, and F. L. Teixeira, "Split-field PML implementations for the unconditionally stable LOD-FDTD method," *IEEE Microw. Wireless Compon. Lett.*, vol. 16, pp. 398-400, 2006.
- [14] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *SIAM J.*, vol. 3, pp. 28-41, 1955.
- [15] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. Amer. Math. Soc.*, vol. 82, pp.421-439, 1956.
- [16] R. Holland, "Implicit three-dimensional finite differencing of Maxwell's equations," *IEEE Trans. Nucl. Sci.*, vol. NS-31, pp. 1322-1326, 1984.
- [17] P. M. Goorjian, "Finite difference time domain algorithm development for Maxwell equations for computational electromagnetism," *Proc. IEEE Antennas and Propagation Society Int. Symp.*, vol. 1, pp. 878-881, 1990.
- [18] F. H. Zheng, Z. Z. Chen, and J. Z. Zhang, "Toward the development of a three-dimensional unconditionally stable finite-difference time-domain method," *IEEE Trans. Microwave Theory Tech.*, vol. 48, pp. 1550-1558, 2000.
- [19] T. Namiki, "3-D ADI-FDTD method-Unconditionally stable time-domain algorithm for solving full vector Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol 48, pp. 1743-1748, 2000.
- [20] C. Gabriel and S. Gabriel, *Compilation of the Dielectric Properties of Body Tissues at RF and Microwave Frequencies, Preprint AL/OE-TR-1996-0037*. San Antonio, Texas: Armstrong Laboratory Brooks AFB, 1996.
- [21] S. Gabriel, R. W. Lau, and C. Gabriel, "The dielectric properties of biological tissues: II. Measurements in the frequency range 10Hz to 20GHz," *Phys. Med. Biol.*, vol. 41, pp. 2251-2269, 1996.
- [22] S. Su, W. Dai, D. T. Haynie, R. Nassar, and N. Simicevic, "Numerical simulation of nanopulse penetration of biological matter using the z-transform," *Journal of Mathematical Modeling and Algorithms*, vol. 4, pp. 99-110, 2005.

- [23] D. M. Sullivan, *Electromagnetic Simulation Using the FDTD Method*. New York, NY: IEEE Press, 1999.
- [24] P. Šolín, *Partial Differential Equations and the Finite Element Method*. Hoboken, NJ: John Wiley & Sons, Inc., 2006.
- [25] N. Ida, *Numerical Modeling for Electromagnetic Non-Destructive Evaluation*. London, U.K.: Chapman & Hall, 1995.
- [26] F. T. Ulaby, *Fundamentals of Applied Electromagnetics*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2006.
- [27] M. N. O. Sadiku, *Elements of Electromagnetics*, 4th ed. New York, NY: Oxford University Press, 2006.
- [28] R. A. Serway and J. W. Jewett, *Physics for Scientists and Engineers with Modern Physics*, 4th ed. Stamford, CT: Cengage Learning, 2009.
- [29] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas and Propagation*, vol. 14, pp. 302-307, 1966.
- [30] A. Taflove and K. R. Umashankar, "Solution of complex electromagnetic penetration and scattering problems in unbounded regions," in *Computational Methods for Infinite Domain Media-Structure Interactions*, vol. 46. Washington, DC: American Society of Mechanical Engineers, 1981, pp. 83-113.
- [31] A. Taflove, "Application of the finite-difference time-domain method to sinusoidal steady-state electromagnetic-penetration problems," *IEEE Trans. EM Comp.*, vol. 22, pp. 191-202, 1980.
- [32] A. Taflove and M. E. Brodwin, "Computation of the electromagnetic fields and induced temperature within a model of the microwave irradiated human eye," *IEEE Micro. Theo. Tech.*, vol. 23, pp. 888-896, 1975.
- [33] A. Taflove and K. R. Umashankar, "A hybrid moment method/finite difference time domain approach to electromagnetic coupling and aperture penetration into complex geometries," *IEEE Trans. Ant. Prop.*, vol. 30, pp. 617-627, 1982.
- [34] K. R. Umashankar and A. Taflove, "A novel method to analyze electromagnetic scattering of complex objects," *IEEE Trans. EM Comp.*, vol. EMC-24, pp. 397-405, 1982.
- [35] A. Taflove and K. R. Umashankar, "The finite difference time domain method for numerical modeling for electromagnetic wave interactions," *Electromagnetics*, vol. 10, pp. 105-126, 1990.

- [36] R. C. Dorf, *The Electrical Engineering Handbook*, 2nd ed. Boca Raton, FL: CRC Press, 1997.
- [37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. New York, NY: Cambridge University Press, 2007.
- [38] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *SIAM J.*, vol. 3, pp. 28-41, 1955.
- [39] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. Amer. Math. Soc.*, vol. 82, pp. 421-439, 1956.
- [40] C. Gabriel, S. Gabriel, and E. Corthout, "The dielectric properties of biological tissues: I. Literature survey," *Phys. Med. Biol.*, vol. 41, pp. 2231-2249, 1996.
- [41] K. R. Foster and H. P. Schwan, "Dielectric properties of tissues and biological materials: A critical review," *Crit. Rev. Biomed. Eng.*, vol. 17, pp. 25-104, 1989.
- [42] W. Kuang and S. O. Nelson, "Dielectric relaxation characteristics of fresh fruits and vegetables from 3 to 20 GHz," *Journal of Microwave Power and Electromagnetic Energy*, vol. 32, pp. 114-122, 1997.
- [43] W. D. Hurt, "Multiterm Debye dispersion relations for permittivity of muscle," *IEEE Trans. Biomed. Eng.*, vol. 32, pp. 60-63, 1985.
- [44] M. Mrozowski and M. A. Stuchly, "Parameterization of media dispersive properties for FDTD," *IEEE Trans. Antennas Propag.*, vol. 45, pp. 1438-1439, 1997.
- [45] D. F. Kelley and R. J. Luebbers, "Debye function expansions of empirical models of complex permittivity for use in FDTD solutions," *IEEE Antennas Propag. Soc. Int. Symp.*, vol. 4, pp.372-375, 2003.
- [46] X. Li and S. C. Hagness, "A confocal microwave imaging algorithm for breast cancer detection," *IEEE Microw. Wireless Compon. Lett.*, vol.11, pp. 130-132, 2001.
- [47] K. S. Cole and R. H. Cole, "Dispersion and absorption in dielectrics I. Alternating current characteristics," *J. Chemical Physics*, vol. 9, pp. 341-351, 1941.
- [48] S. Gabriel, R. W. Lau, and C. Gabriel, "The dielectric properties of biological tissues: III. Parametric models for the dielectric spectrum of tissues," *Phys. Med. Biol.*, vol. 41, pp. 2271-2293, 1996.

- [49] S. Su, "Numerical simulation of nanopulse penetration of biological matter using the z-transform," Ph.D dissertation, Louisiana Tech Univeristy, Ruston, LA, 2005.
- [50] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic field equations," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-23, pp. 377–382, 1981.
- [51] Z. P. Liao, H. L. Wong, B. P. Yang, and Y. F. Yuan, "A transmitting boundary for transient wave analysis," *Scientia Sinica*, vol. 27, pp. 1063–1076, 1984.
- [52] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, vol. 114, pp. 185–200, 1994.
- [53] J. P. Berenger, "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves," *J. Comp.Phys.*, vol. 127, pp. 363-379, 1996.
- [54] S. D. Gedney, "An anisotropic perfectly matched layer absorbing media for the truncation of FDTD lattices," *IEEE Transactions on Antennas and Propagation*, vol. 44, pp. 1630–1639, 1996.
- [55] D. K. Cheng, *Field and Wave Electromagnetics*. Menlo Park, CA: Addison Wesley, 1992.
- [56] D. M. Sullivan, "A simplified PML for use with the FDTD method," *IEEE Microwave and Guided Wave Letters*, vol. 6, pp. 97-99. 1996.
- [57] D. M. Sullivan, "An unsplit step 3-D PML for use with the FDTD method", *IEEE Microwave and Guided Wave Letters*, vol. 7, pp. 184-186, 1997.
- [58] Z. S. Sacks, D. M. Kingsland, R. Lee, and J.F. Lee, "A perfectly matched anisotropic absorber for use as an absorbing boundary condition," *IEEE Trans. Antennas Propag.*, vol. 43, pp. 1460-1463, 1995.
- [59] S. D. Gedney, *Introduction to the Finite-Difference Time-Domain (FDTD) Method for Electromagnetics*. San Rafael, CA: Morgan & Claypool Publishers, 2011.
- [60] U. S. Inan and R. A. Marshall, *Numerical Electromagnetics: The FDTD Method*. New York, NY: Cambridge University Press, 2011.
- [61] J. Jin, *Theory and Computation of Electromagnetic Fields*. Hoboken, NJ: John Wiley & Sons. Inc., 2010.