Development and extensions of stable and efficient alternating-direction-implicit / locally one-dimensional FDTD methods

Tay, Wei Choon

2014

Tay, W. C. (2014). Development and extensions of stable and efficient alternating-direction-implicit / locally one-dimensional FDTD methods. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61816>

<https://doi.org/10.32657/10356/61816>

**DEVELOPMENT AND EXTENSIONS OF
STABLE AND EFFICIENT
ALTERNATING-DIRECTION-IMPLICIT /
LOCALLY ONE-DIMENSIONAL FDTD
METHODS**

TAY WEI CHOON

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2014

Acknowledgements

I would like to express my gratitude to my academic supervisor and mentor, Assoc. Prof. Tan Eng Leong for his guidance and support throughout my Ph.D. candidature. His passion and commitment has motivated me to reach my utmost accomplishment towards professional maturity.

I would also like to thank Nanyang Technological University, Singapore and the faculty, School of Electrical and Electronic Engineering for providing me with all the research facilities and funding of my scholarship.

Last but not least, a special thanks to my parents and family members for their love and support. This journey would not have been possible with their love and support.

Table of Contents

Acknowledgements	i
Abstract	vi
List of Abbreviations and Symbols	ix
List of Figures	xii
List of Tables	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	4
1.3 Contributions	4
1.4 Organization	7
2 Literature Survey	10
2.1 FDTD Methods for Electromagnetics	10
2.1.1 Yee's Explicit FDTD Method	11
2.1.2 ADI-FDTD Method	15
2.1.3 LOD-FDTD Method	20
2.1.4 Graphics Processing Units	23
2.1.5 Absorbing Boundary Conditions	25
2.2 FDTD Methods for Heat Transfer	31

2.2.1	Explicit Method	33
2.2.2	Implicit Methods	34
2.2.3	Douglas-Gunn (DG) ADI Method	35
2.3	FDTD Methods for Schrödinger Equation	37
2.3.1	Complex Explicit Method	39
2.3.2	Staggered Explicit Method	40
2.3.3	Tridiagonal ADI (Tri-ADI) Method	42
3	Development of Stable and Efficient ADI/LOD-FDTD Methods for Electromagnetics	44
3.1	Introduction	44
3.2	Efficient Fundamental ADI- and LOD-FDTD Methods	47
3.2.1	FADI-FDTD Method	47
3.2.2	FLOD-FDTD Method	54
3.3	PMC and PEC Boundary Conditions for FDTD Methods	58
3.3.1	Boundary conditions for FADI-FDTD Method	58
3.3.2	Boundary conditions for FLOD-FDTD Method	60
3.3.3	Numerical Results	62
3.4	Mur ABC for FDTD Methods	65
3.4.1	Mur ABC for ADI- and LOD-FDTD Methods	65
3.4.2	Mur ABC for FADI-FDTD Method	69
3.4.3	Mur ABC for FLOD-FDTD Method	74
3.4.4	Numerical Results	77
3.5	Split-Field PML for ADI-FDTD Method	83
3.5.1	ADI-FDTD method with Split-field PML	84
3.5.2	FADI-FDTD Method with Split-Field PML	86
3.5.3	Numerical Results	90
3.6	CFS-CPML for ADI-FDTD Method	92
3.6.1	ADI-FDTD Method with CFS-CPML	93
3.6.2	FADI-FDTD Method with CFS-CPML	96

3.6.3	GPU-accelerated FADI-FDTD method with CFS-CPML . . .	100
3.6.4	Numerical Results	102
3.6.5	Application of FADI-FDTD method with CFS-CPML	106
3.7	Conclusions	109
4	Extension of Stable and Efficient ADI/LOD-FDTD Methods for Heat Transfer	111
4.1	Introduction	111
4.2	2-D ADI Method	115
4.2.1	Proposed Stabilized DG-ADI and PR-ADI Methods	117
4.2.2	Stable and Efficient FADI Method	123
4.2.3	Stability Analysis, Efficiency and Numerical Results	126
4.3	2-D ADI Method For ICs with microchannel cooling	135
4.3.1	FADI method for ICs with microchannel cooling	135
4.3.2	GPU-accelerated FADI method for ICs with microchannel cooling	138
4.3.3	Numerical Results	140
4.4	3-D ADI Method	144
4.4.1	Efficient ADI method	144
4.4.2	Memory Allocation	148
4.4.3	Boundary Conditions	149
4.4.4	Numerical Experiments	153
4.5	3-D LOD Method	156
4.5.1	Potentially Unstable DG-ADI method with gradient terms . .	157
4.5.2	Proposed Stable LOD method with gradient terms	158
4.5.3	Potentially Unstable DG-ADI method without gradient terms	161
4.5.4	Proposed Stable LOD method without gradient terms	162
4.5.5	Stable and Efficient FLOD Method	164
4.5.6	Memory Allocation	166
4.5.7	Boundary Conditions	168

4.5.8	Stability Analysis, Efficiency and Numerical Results	169
4.6	Conclusions	180
5	Extension of Stable and Efficient ADI-FDTD Method for Schrödinger Equation	183
5.1	Introduction	183
5.2	Pentadiagonal ADI (Penta-ADI) Method	186
5.3	Pentadiagonal FADI (Penta-FADI) Method	188
5.4	Boundary Conditions	193
5.5	Numerical Experiments	196
5.5.1	Example 1	196
5.5.2	Example 2	200
5.5.3	Efficiency	204
5.6	Conclusions	205
6	Conclusions and Future Work	206
6.1	Conclusions	206
6.2	Future Work	209
	Appendix	210
	Author's Publications	245
	Bibliography	247

Abstract

Recently, there has been increasing interest in the development of unconditionally stable finite-difference time-domain (FDTD) methods in electromagnetics that are not constrained by the Courant-Friedrich-Lewy (CFL) condition. This thesis presents the development and extensions of stable and efficient alternating-direction-implicit (ADI) or locally one-dimensional (LOD) FDTD methods. From Maxwell's equations (which are hyperbolic partial differential equations), the efficient fundamental ADI (FADI) and LOD (FLOD) FDTD methods are formulated into simpler, more concise, and more efficient form of the ADI- and LOD-FDTD methods. The boundary conditions are first investigated on the FADI- and FLOD-FDTD methods. For closed region simulation, the perfect magnetic conductor (PMC) and perfect electric conductor (PEC) are derived regardless of the implicit updating of electric or magnetic fields. Next, for unbounded region simulation, the Mur absorbing boundary condition (ABC) is incorporated into the FADI- and FLOD-FDTD methods using consistent implementation and a novel implementation with lower reflection coefficient. For even better absorption, the perfectly matched layers (PMLs) have been incorporated into the FADI-FDTD method. Further, the FADI-FDTD method with complex frequency shifted convolutional PML (CFS-CPML) has been incorporated into the graphics processor units (GPUs) to exploit data parallelism for higher efficiency. To demonstrate the usefulness of the FADI-FDTD method with CFS-CPML, a practical microstrip low-pass filter is presented. A high computational power is attained while preserving a good agreement with the Yee-FDTD

method.

Based on the FDTD technique used for solving Maxwell's equation in electromagnetics, it can be extended to solve the heat transfer equation (which is a parabolic partial differential equation) in thermodynamics. The thesis next proposes the stable and efficient FADI and FLOD methods for heat transfer. In inhomogeneous media, the two-dimensional (2-D) Douglas-Gunn (DG) ADI method for the heat transfer equation takes into account both Laplacian and gradient terms. The potential instability of the conventional DG-ADI method caused by the gradient terms is alleviated. Subsequently, the proposed stabilized DG-ADI method is cast into the (stabilized) Peaceman-Rachford (PR) ADI method in compact form. A stable and efficient FADI method is then formulated. The FADI method is further extended into the heat transfer equation with convection heat flux due to fluid motion. To achieve higher efficiency, the FADI method has been incorporated into the GPU for efficient thermal simulation of integrated circuits (ICs) with microchannel cooling. For the three-dimensional (3-D) DG-ADI method in the homogeneous media, its implementation is highly complex and requires a considerable number of memory variables. To overcome these complications, the conventional DG-ADI method is formulated into the efficient ADI method with single operator and heat generation input on the right-hand-side (RHS) of the first procedure. The proposed method has substantially less RHS update coefficients and field variables; as well as lower number of memory variables than the conventional DG-ADI method. Thus, it achieves higher efficiency with reduced memory indexing and arithmetic operations. As the current 3-D DG-ADI method is still conditionally stable within inhomogeneous media, two stable 3-D FLOD methods are proposed for solving the heat transfer equation. Stability analysis by means of analyzing the eigenvalues of the amplification matrix is provided to substantiate the stability of the FLOD method. Further, the relative maximum error of the FLOD method for heat transfer is ex-

amined. It exhibits good trade-off between accuracy and efficiency. To show the effectiveness of the proposed stabilized FLOD method, the heat distribution of the closely resembled Alpha 21364 processor chip is presented and analyzed.

By using the FDTD technique for solving the heat transfer equation, this technique can be extended to solve the Schrödinger equation (which is a parabolic partial differential equation with complex variables) in quantum mechanics. A stable and efficient FADI method for Schrödinger equation is proposed. For the tridiagonal system equations of the ADI (Tri-ADI) method, the computation of the complex wave function is rather taxing and time consuming. A novel pentadiagonal system of equations for the ADI (Penta-ADI) method is introduced through the separation of the complex wave function into real and imaginary parts. Subsequently, the Penta-ADI method is formulated into the pentadiagonal efficient fundamental ADI (Penta-FADI) method. Such efficient fundamental scheme has matrix-operator-free RHS, leading to computationally efficient update equations. As the Penta-FADI method involves five stencils on the left-hand-sides (LHS) of the pentadiagonal update equations, special treatments are provided for the implementation of the Dirichlet boundary condition. Further analysis of the Penta-FADI method over Tri-ADI method is also shown. Using the Penta-FADI method, a significantly higher efficiency gain can be achieved.

List of Abbreviations and Symbols

Abbreviations

1-D	one-dimensional
2-D	two-dimensional
3-D	three-dimensional
A/S	additions/subtractions
ABC	absorbing boundary condition
ADI	alternating-direction-implicit
BC	boundary condition
CFL	Courant-Friedrich-Lewy
CFLN	Courant-Friedrich-Lewy number
CFS	complex frequency shifted
CN	Crank-Nicolson
CPML	convolutional perfectly matched layer
CPU	central processing unit
CUDA	compute unified device architecture
DG	Douglas-Gunn
FADI	fundamental alternating-direction-implicit
FDTD	finite-difference time-domain
FLOD	fundamental locally one-dimensional
flops	floating-point operations
GPU	graphic processing unit
IC	integrated circuit

LHS	left-hand-side
LOD	locally one-dimensional
M/D	multiplications/divisions
PEC	perfect electric conductor
PEMC	perfect electromagnetic conductor
PMC	perfect magnetic conductor
PML	perfectly matched layer
PR	Peaceman-Rachford
RAM	random access memory
RHS	right-hand-side
SIMD	single-instruction multiple-data
SM	streaming multiprocessor
SS	split-step
TE	transverse electric
TM	transverse magnetic

Symbols

E_x, E_y, E_z	electric field components
H_x, H_y, H_z	magnetic field components
H_{zx}, H_{zy}	magnetic field subcomponents
t, f, ω	time, frequency, angular frequency
$\Delta x, \Delta y, \Delta z, \Delta t$	spatial and time steps
ϵ, μ	permittivity, permeability
$\sigma_x, \sigma_y, \sigma_z$	electric conductivities in PML region
$\sigma_x^m, \sigma_y^m, \sigma_z^m$	magnetic conductivities in PML region
σ, σ^m	electric and magnetic conductivities
$T, T_{\text{DG}}, T_{\text{PR}},$	time-dependent temperature

$T_{\text{EFF}}, T_{\text{LOD}}, T_{\text{F}}$	
h_x, h_y, h_z	heat transfer coefficients
κ, ρ, C_p	thermal conductivity, density, specific heat capacity
g, T_{∞}	heat energy generation rate, pre-specified temperature
$\vec{u}, u_x, u_y$	fluid velocity vector and components
q	specific heat flux
$\square_z, \square_z, \square_z$	spatial averaging operators
Ψ, Ψ_R, Ψ_I	complex, real and imaginary Schrödinger wave function
$\hat{H}, p$	Hamiltonian operator, momentum operator
$\hat{K}, \hat{\omega}$	kinetic energy operator, potential energy function
$\hbar, m_e$	reduced Planck's constant, particle mass

List of Figures

2.1	Yee's lattice cell.	12
2.2	Field components (a) within a quarter of a cell, (b) in a plane.	13
2.3	Device architecture.	24
2.4	Execution of a Typical CUDA Program.	25
2.5	Structure of a 2-D TE, FDTD grid employing the Berenger PML ABC in [39].	28
3.1	Geometry with the shaded region representing either PMC or PEC boundary conditions.	62
3.2	Incident and reflected waveforms for (a) PMC boundary condition and (b) PEC boundary condition.	63
3.3	The reflection coefficients of the Mur ABC for (a) consistent implementation and (b) novel implementation in the conventional and efficient fundamental 3-D ADI-FDTD methods with various CFLN.	78
3.4	The reflection coefficients of the Mur ABC for (a) consistent implementation and (b) novel implementation in the conventional and efficient fundamental 3-D LOD-FDTD methods with various CFLN.	79
3.5	Reflection errors for the conventional and efficient fundamental ADI-FDTD methods with split-field PML incorporated for various CFLN.	91
3.6	CUDA Threads Organization for E_x	100
3.7	Flowchart of the FADI-FDTD method with CFS-CPML as implemented on GPU.	101
3.8	Comparison of time domain waveforms for the electric field at observation point (CFLN = 4).	103
3.9	Reflection errors of CPU and GPU-accelerated FADI-FDTD methods with CFS-CPML for (a) CFLN = 1, (b) CFLN = 1 (magnified), (c) CFLN = 2 and (d) CFLN = 4.	104

3.10	Microstrip low-pass filter [109].	107
3.11	Return loss of the low-pass filter with various CFLNs.	108
3.12	Insertion loss of the low-pass filter with various CFLNs.	108
4.1	Heat transfer simulation layout of a piecewise homogeneous medium.	127
4.2	Transient temperature at observation point ($i = 60, j = 60$) computed using the conventional DG-ADI method [c.f. (4.2.4)], $\gamma = 5$	128
4.3	Transient temperature at observation point ($i = 60, j = 60$) computed using the FADI method [c.f. (4.2.30)] for various γ	128
4.4	Scatter plot of eigenvalues of the reduced amplification matrix for the conventional DG-ADI method [c.f. (4.2.4)] for various γ	130
4.5	Scatter plot of eigenvalues of the reduced amplification matrix for the FADI method [c.f. (4.2.30)] for various γ	131
4.6	Transient temperature at observation point ($i = 19, j = 21$), computed by (a) the conventional DG-ADI method [c.f. (4.2.4)] (unstable) and (b) the FADI method [c.f. (4.2.30)] (stable).	133
4.7	Scatter plot of eigenvalues of the reduced amplification matrix for (a) conventional DG-ADI method [c.f. (4.2.4)] and (b) FADI method [c.f. (4.2.30)].	134
4.8	Flowchart of the GPU-accelerated FADI method for thermal simulation.	138
4.9	Five-layer silicon structure.	140
4.10	Power density of the functional blocks layer.	141
4.11	Temperature profile of the functional blocks layer at 0.2 s.	141
4.12	Transient temperature results at observation point ($i = 170, j = 120, k = 4$) with various γ	142
4.13	Transient temperature results at observation point ($i = 170, j = 120, k = 4$) from zero to eight microchannels.	143
4.14	Pseudocode of 3-D efficient ADI method across iterations.	148
4.15	Pseudocode of 3-D DG-ADI method across iterations.	149
4.16	Pseudocode of 3-D DG-ADI method in main grid for the first procedure of iteration.	150
4.17	Pseudocode of 3-D DG-ADI method in $x - z$ plane at $y = 0$ for the first procedure of iteration.	151

4.18	Pseudocode of the 3-D efficient ADI method in main grid for the first procedure of iteration.	152
4.19	Pseudocode of 3-D efficient ADI method in $x - z$ plane at $y = 0$ for the first procedure of iteration.	153
4.20	A chip layout with the power density in each hierarchical function block.	154
4.21	Temperature profile of the chip at 0.12 s.	154
4.22	Transient temperature results using DG-ADI and efficient ADI methods with various γ	155
4.23	Pseudocode of 3-D FLOD method across iterations.	167
4.24	Pseudocode of 3-D LOD method across iterations.	167
4.25	Pseudocode of 3-D FLOD method for the first procedure of iteration.	168
4.26	Pseudocode of 3-D LOD method for the first procedure of iteration.	169
4.27	Layout of a computation domain with piecewise homogeneous medium.	170
4.28	Scatter plot of eigenvalues of the amplification matrix for 3-D DG-ADI method [c.f. (4.5.8)] for various γ	171
4.29	Scatter plot of eigenvalues of the amplification matrix for 3-D FLOD method [c.f. (4.5.15)] for various γ	172
4.30	Transient temperature at observation point ($i = 4, j = 4, k = 4$) computed using 3-D DG-ADI method [c.f. (4.5.8)], $\gamma = 30$	173
4.31	Transient temperature at observation point ($i = 4, j = 4, k = 4$) computed using 3-D FLOD method [c.f. (4.5.15)], for various γ	173
4.32	A chip layout with the power density in each hierarchical function block.	175
4.33	Temperature profile of the chip at 0.1 s.	176
4.34	Transient temperature results using 3-D FLOD method at observation point ($i = 60, j = 60, k = 12$) with various γ	176
4.35	Relative maximum error for 3-D FLOD method with various γ	177
4.36	Floorplan of Alpha 21364 processor.	178
4.37	Power density in each hierarchical function block.	179
4.38	Temperature profile of the chip in steady state.	179
5.1	Computation domain of 11×11 grids.	193

5.2	Waveform at observation point (31,51) for Example 1.	197
5.3	Probability of Example 1.	200
5.4	Waveform of $\Psi_R _{250,250}$ for Example 2.	201
5.5	Waveform of $\Psi_I _{250,250}$ for Example 2.	201
5.6	Probability of Example 2.	203

List of Tables

3.1	Flops count for the conventional and efficient fundamental ADI-FDTD methods	53
3.2	Flops count for the conventional and efficient fundamental LOD-FDTD methods	57
3.3	Flops count for the PMC and PEC boundary equations	64
3.4	Flops count of consistent and novel implementations for the Mur ABC	81
3.5	CPU time and average reflection coefficient for (a) consistent implementation and (b) novel implementation of the conventional and efficient fundamental ADI schemes	82
3.6	CPU time and average reflection coefficient for (a) consistent implementation and (b) novel implementation of the conventional and efficient fundamental LOD schemes	83
3.7	Flops count for the split-field PML using conventional and efficient fundamental ADI-FDTD methods	92
3.8	Flops count for the CFS-PML using conventional and efficient fundamental FADI-FDTD methods	99
3.9	Comparison of CPU and GPU computation time for FADI-FDTD and Yee-FDTD methods with CFS-CPML (For the FADI-FDTD, CFLN = 4, 250 time steps. For the Yee-FDTD, CFLN = 1, 1000 time steps)	106
4.1	Efficiency gains of the FADI method over PR-ADI and DG-ADI methods for various computation domains ($\gamma = 5$, 3000 time steps)	132
4.2	Efficiency gains of the (GPU) FADI method over (CPU) PR-ADI, DG-ADI and explicit methods for various computation domains ($\gamma = 50$, 3×10^4 time steps)	139
4.3	Efficiency gains of the efficient ADI method over DG-ADI method for various computation domain ($\gamma=2$, 3000 time steps)	156

4.4	Efficiency gains of the FLOD method over LOD, DG-ADI and explicit methods for various computation domains ($\gamma=50$)	174
4.5	CPU time and relative maximum error of the FLOD method	177
5.1	Maximum relative error for the Penta-FADI method at different times for Example 1 with $\Delta x = \Delta y = 0.05$ m and $\Delta t = 6.25 \times 10^{-4}$ s. . . .	198
5.2	Maximum relative error for the Penta-FADI method at time $T = 1$ s for Example 1 with different Δx and Δy where $\Delta t = 6.25 \times 10^{-4}$ s. . .	199
5.3	Maximum relative error for the Penta-FADI method at different times for Example 2 with $\Delta x = \Delta y = 0.1$ m and $\Delta t = 1.0 \times 10^{-3}$ s. . . .	202
5.4	CPU time and relative maximum error of the Penta-FADI method with $\Delta x = \Delta y = 0.1$ m and $\Delta t = 1.0 \times 10^{-3}$ s.	202
5.5	Efficiency gains of Tri- and Penta-FADI methods over Tri-ADI method for various computation domains ($\Delta t = 4.0 \times 10^{-3}$ s, 250 time steps) . . .	204

Chapter 1

Introduction

1.1 Motivation

Computational electromagnetics encompasses the modeling, simulation and analysis of the interaction among electromagnetic fields with physical objects and the environment. It typically involves using computationally efficient approximations to Maxwell's equations and is used to calculate antenna performance, electromagnetic compatibility, radar cross section and electromagnetic wave propagation when not in free space. There are two major types of methods in computational electromagnetics: (1) frequency-domain methods such as Method of Moments (MoM) and finite element method (FEM); (2) time-domain methods such as finite-difference method.

The finite-difference method is a numerical method for solving partial differential equations. Using the finite-difference equations, approximation of solutions can be attained by the derivatives of partial differential equations. By incorporating the finite-difference method to the time-domain version, the finite-difference time-domain (FDTD) method has flourished to become a popular method to model many problems dealing electromagnetic wave interactions. To describe some of them, FDTD modeling applications can be found in areas such as wireless communications devices, biomedical imaging/treatment, nanoplasmonics etc. The FDTD method al-

allows a single simulation with a broadband pulse used as the source to generate a wide range of frequencies for the response of the system. This method is known to be more flexible than other approaches such as FEM for handling complex structures. With technological advancement, high speed parallel-processing computers have dominated supercomputing. The FDTD methods can be incorporated easily to the parallel-processing central processing unit (CPU) based computers with high efficiency, and scale extremely well on recently developed programmable graphics processor unit (GPU) with highly parallel processors.

The FDTD method can also be used in other branches of physics. In computational thermodynamics, the heat transfer equation is a parabolic partial differential equation used to describe the distribution of heat in a given region over time. Based on the FDTD technique used for solving Maxwell's equation in electromagnetics, it has been extended to solve the heat transfer equation for heat analysis of electrical devices such as IC chips. Using this time-domain technique, the FDTD method has become increasingly useful in the area of heat transfer to analyze heat distribution in transient state as well as steady state, ensuring device operability and reliability. The analysis of the thermal distribution for complicated packaging systems such as three-dimensional (3-D) die stacking in embedded processor systems which increases significantly with technological advancement is of crucial importance. Another branch of physics of great importance is quantum mechanics. Quantum effects of lasers, transistors, electron microscope and magnetic resonance imaging (MRI) are indispensable parts of modern electronics systems and devices. The FDTD methods can be used in the Schrödinger equation which is a parabolic partial differential equation with complex variables. Using the FDTD technique for solving the heat transfer equation, this technique can be extended to solve the Schrödinger equation in quantum mechanics. Through the Schrödinger equation, the behavior in the systems at atomic length scales and smaller can be analyzed.

The conventional explicit FDTD method has a stability constraint such that its time step size is restricted by the Courant-Friedrich-Lewy (CFL) condition. This imposes a maximum limit on the chosen time step depending on the minimum spatial step of the computation domain. To overcome the stability constraint, unconditionally stable alternating-direction-implicit (ADI) and locally one-dimensional (LOD) FDTD methods have been proposed. These methods attract wide interest because the time step is no longer restricted by the CFL condition. However, the ADI- and LOD-FDTD methods come with some shortcomings such as increased complexity in implementation. Besides having to solve the tridiagonal system of equations, there are substantial amount of arithmetic operations and field variables involved on the right-hand-sides (RHS) of the update equations, not to mention the huge amount of memory indexing operations incurred. These in turn increase the programming complexity and CPU computation time.

In electromagnetics, the incorporation of various boundary conditions for closed region or unbounded region problem into ADI- and LOD-FDTD methods further complicate the already complex updating equations. With the inclusion of boundary parameters, the amount of aforementioned field variables, arithmetic and memory indexing operations have all increased considerably. This results in further degradation of the overall efficiency. It is therefore hopeful that the computational complexity and memory overheads due to the incorporation of various boundary conditions can be addressed. For the analysis of heat transfer using FDTD methods, the heat transfer equation is also restricted by stability constraints arising from both Laplacian and gradient terms. The application of the explicit FDTD method becomes increasingly difficult especially when one or both of the constraints become too restrictive. Furthermore, the current ADI-FDTD method only removes the stability constraint for the Laplacian terms. The potential instability caused by the gradient terms is yet to be resolved. To describe the changes in quantum state of physical

system over time, the Schrödinger equation has been introduced. Merely applying the FDTD method for the Schrödinger equation is not desirable as the nature of the Schrödinger equation involves complex wave function which is computationally expensive. Therefore, there is a need to alleviate the complex nature of the Schrödinger equation so as to achieve high computational efficiency.

1.2 Objectives

Motivated by the observations outlined in the preceding section, the objectives of the thesis are as follows:

- 1) To develop stable and efficient ADI- and LOD-FDTD methods for electromagnetics. Thereafter, to study the incorporation of various boundary conditions into the implicit FDTD methods and hence, to improve on their accuracy and efficiency.
- 2) To extend the ADI-FDTD method for heat transfer. Subsequently, to propose a stable and efficient implicit FDTD method for heat transfer.
- 3) To extend the ADI-FDTD method for the Schrödinger equation. In addition, to alleviate the complex nature of the Schrödinger equation so as to improve the computational efficiency.

1.3 Contributions

In this thesis, we present our development and extensions of stable and efficient ADI- or LOD-FDTD methods, as well as numerical results to substantiate our approach. The following summary lists the primary contributions of this work.

- 1) In electromagnetics, various boundary conditions for the conventional ADI- and LOD-FDTD methods have been investigated and formulated into the efficient fun-

damental ADI (FADI) and LOD (FLOD) FDTD methods. For closed region simulation, the perfect magnetic conductor (PMC) and perfect electric conductor (PEC) boundary conditions have been derived and implemented into the FADI- and FLOD-FDTD methods regardless of whether the update equations for electric or magnetic fields are implicit. Comparisons between the PMC and PEC boundary equations in the conventional and efficient fundamental schemes of ADI- and LOD-FDTD signify a reduction in the floating-point operations (flops) count, and hence justifying the better efficiency and simplicity for the efficient fundamental schemes.

2) To simulate unbounded region problem, the Mur first order absorbing boundary condition (ABC) has been incorporated into the FADI- and FLOD-FDTD methods using consistent implementation and a novel implementation with lower reflection coefficient. By comparing the CPU computation time of both conventional and efficient fundamental ADI- and LOD-FDTD methods, it is ascertained that substantial gain in the overall efficiency has been achieved for the latter even with the incorporation of the Mur ABC. For better absorption and higher efficiency, the original split-field perfectly matched layer (PML) and complex frequency shifted convolutional PML (CFS-CPML) have been formulated into the FADI-FDTD method. The FADI-FDTD method with different PMLs incorporated has been compared and validated with the conventional ADI-FDTD implementation through numerical simulations. Furthermore, by using Compute Unified Device Architecture (CUDA), the FADI-FDTD method with CFS-CPML can be incorporated into the GPU to exploit of data parallelism, and hence achieving a greater computational efficiency. To demonstrate the usefulness of the FADI-FDTD method with CFS-CPML, a practical microstrip low-pass filter has been presented. A high computational power has been attained while preserving a good agreement with the Yee-FDTD method.

3) For heat transfer, a stabilized two-dimensional (2-D) Douglas-Gunn (DG) ADI

method has been proposed to solve the heat transfer equation which takes into consideration both Laplacian and gradient terms. This proposed method is then cast into the (stabilized) Peaceman-Rachford (PR) ADI method in compact form, and further formulated into the FADI method with operator-free RHS, resulting in simpler and more concise update equations. The relationships among temperatures resulted from these three methods, namely, DG-ADI, PR-ADI and FADI methods have been presented and discussed. Stability analysis by means of analyzing the eigenvalues of the reduced amplification matrix has been performed to verify the stability of the FADI method. Subsequently, the convection heat flux due to fluid motion has been included in the heat transfer equation. Using the proposed 2-D FADI method, the efficient thermal simulation of integrated circuits (ICs) with microchannel cooling has been presented. To further accelerate the FADI method, the GPU has been utilized through CUDA implementation to achieve higher efficiency gain.

4) The conventional 3-D DG-ADI method in the homogeneous media has been formulated into the 3-D efficient ADI method with single operator and heat generation input on the RHS of the first procedure, therefore reducing the number of arithmetic operations to the minimal. Using the efficient ADI method, there has been a decrease in the number of memory variables required, which further reduces the memory space and memory indexing overhead. Due to the potential instability of the 3-D DG-ADI method within inhomogeneous media, two stable 3-D LOD methods have been proposed to solve the heat transfer equation. The first stable LOD method for solving the heat transfer equation takes into account both Laplacian and gradient terms. On the other hand, the second stable LOD method involves the discretization of the heat transfer equation with gradient terms being absorbed into the finite-difference operator directly without being expanded out. The stable LOD method is then cast into the compact form and formulated into the FLOD method

with operators-free RHS. Stability analysis by means of analyzing the eigenvalues of the amplification matrix substantiates the stability of the FLOD method. The potential instability of the conventional DG-ADI method for inhomogeneous media has also been demonstrated. Furthermore, the relative maximum error of the FLOD method has been examined and illustrated, which exhibits good trade-off between accuracy and efficiency. To show the effectiveness of the FLOD method, the heat distribution of the closely resembled Alpha 21364 processor chip has been presented and analyzed.

5) For the Schrödinger equation, through the separation of the complex wave function into real and imaginary parts, a novel pentadiagonal system of equations for the ADI (Penta-ADI) method has been attained. The proposed approach is further formulated into the Pentadiagonal FADI (Penta-FADI) method with matrix-operator-free RHS, resulting in computationally efficient updating equations. As the Penta-FADI method involves five stencils on the left-hand-side (LHS) of pentadiagonal update equations, special treatments that are required for the implementation of the Dirichlet boundary condition have been discussed and presented. Numerical results substantiate the efficiency gain of the Penta-FADI method over Tri-ADI method, which involves the computations of the complex wave function.

1.4 Organization

This thesis consists of six chapters. The remaining chapters are as follows:

Chapter 2 reviews the past literature for related works on FDTD methods. It reviews and describes the conventional Yee's explicit FDTD method as well as some other unconditionally stable methods for Maxwell's equations which form the basis of electromagnetic theory, together with some of its properties and its relation to current work. Various ABCs for simulating electromagnetic waves propagating con-

tinuously beyond the computational space to simulate open structures are detailed. The explicit method and various unconditionally stable implicit methods for heat transfer which analyze thermal effects caused by power distribution and dissipation are discussed. Lastly, two explicit quantum FDTD methods and the unconditionally stable Tri-ADI method for the Schrödinger equation which forms the basis of quantum mechanics are reviewed.

Chapter 3 presents the development of stable and efficient ADI/LOD-FDTD methods for electromagnetics. We demonstrate the formulation of the conventional ADI- and LOD-FDTD methods into their respective efficient fundamental forms. Subsequently, the detailed implementations of both PMC and PEC boundary conditions for FADI- and FLOD-FDTD methods for closed region simulation are presented. For open structure simulation, the consistent and novel implementations of the Mur first order ABC in 3-D FADI- and FLOD-FDTD methods are described and their reflection coefficients are investigated. To attain better absorption, the formulations of the original split-field PML and CFS-CPML for the FADI-FDTD method are presented. To achieve a greater computational efficiency, we further incorporate the FADI-FDTD method with CFS-CPML into the GPUs for the exploit of data parallelism. Numerical results of reflection error and efficiency gain are discussed. To demonstrate the usefulness of the FADI-FDTD method with CFS-CPML, a practical microstrip low-pass filter is presented.

Chapter 4 presents the extension of stable and efficient ADI/LOD-FDTD methods for heat transfer. A stable 2-D FADI method for solving the heat transfer equation with both Laplacian and gradient terms is presented. By means of analyzing the eigenvalues of the amplification matrix, the stability of the FADI method is investigated. The proposed stabilized 2-D FADI method is further incorporated into the GPU to demonstrate the efficient thermal simulation of ICs with microchannel cool-

ing. This is followed by the formulation of the conventional 3-D DG-ADI method into 3-D efficient ADI method. The memory allocation and boundary condition implementation for both 3-D DG-ADI and efficient ADI methods are discussed. Due to the potential instability of 3-D DG-ADI method within inhomogeneous media, two stable 3-D FLOD methods to solve the heat transfer equation are presented. Using the FLOD method, the heat distribution of the closely resembled Alpha 21364 processor chip is illustrated and analyzed.

Chapter 5 presents the extension of stable and efficient ADI-FDTD method for the Schrödinger equation. We proposed a Penta-FADI method which consists of a pentadiagonal system of equations. As the Penta-FADI method involves five stencils on the LHS of pentadiagonal update equations, special treatments that are required for the implementation of the Dirichlet boundary condition are illustrated. Numerical results on the relative maximum relative error and computational efficiency are demonstrated and discussed.

Chapter 6 summarizes the main contributions of this thesis, and lists several recommendations for future research. Finally, the complete update equations for various techniques are included in Appendix.

Chapter 2

Literature Survey

This chapter gives an overview of literature survey done on various areas related to the FDTD methods. The conventional Yee's explicit FDTD method as well as some other unconditionally stable methods for Maxwell's equations which form the basis of electromagnetic theory are reviewed and described. Next, the recent trends of using GPUs for parallel computing are mentioned. Various ABCs for simulating electromagnetic waves propagating continuously beyond the computational space to simulate open structures are detailed. The explicit method and various unconditionally stable implicit methods for heat transfer which analyze thermal effects caused by power distribution and dissipation are discussed. Lastly, two explicit quantum FDTD methods and the unconditionally stable Tri-ADI method for the Schrödinger equation which forms the basis of quantum mechanics are reviewed.

2.1 FDTD Methods for Electromagnetics

The Maxwell's equations are a set of hyperbolic partial differential equations which form the basis of electromagnetic theory. The time-dependent Maxwell's equations in a source-free, lossless, isotropic and non-dispersive medium are given as

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t} \quad (2.1.1a)$$

$$\nabla \times \mathbf{H} = \epsilon \frac{\partial \mathbf{E}}{\partial t} \quad (2.1.1b)$$

with the following symbols definition and units:

$\mathbf{E}$: electric field vector (V/m)

$\mathbf{H}$: magnetic field vector (A/m)

ϵ : electric permittivity (F/m)

μ : magnetic permeability (H/m)

By expanding (2.1.1a)-(2.1.1b), we can obtain six coupled partial differential equations:

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon} \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \quad (2.1.2a)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon} \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \quad (2.1.2b)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \quad (2.1.2c)$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu} \left(\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \right) \quad (2.1.2d)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu} \left(\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \right) \quad (2.1.2e)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu} \left(\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \right) \quad (2.1.2f)$$

The system of six coupled partial differential equations (2.1.2a)-(2.1.2f) form the basis of the FDTD methods in solving various electromagnetics problems in 3-D space.

2.1.1 Yee's Explicit FDTD Method

The Yee's explicit FDTD method was introduced by Kane Yee in 1966 [1], which uses a set of finite-difference equations to discretize Maxwell's curl equations for

Figure 2.1: Yee's lattice cell.

lossless media. In the Yee's explicit FDTD method, the electric and magnetic fields are staggered in the Yee's lattice cell as shown in Figure 2.1.

The E and H components in the 3-D space of the unit cell are placed such that every E component is surrounded by 4 circulating H components, while every H component is surrounded by 4 circulating E components. A simple relationship between the field components is represented in Figure 2.2.

The Yee's finite-difference equations are central difference in nature and second order accurate. Incorporated with the positioning of E and H components in the Yee's lattice unit cell, it automatically enforces the Gauss' Law relations. Therefore, the Yee's lattice unit cell electric and magnetic fields are divergence free in the absence of free electric and magnetic charges.

Figure 2.2: Field components (a) within a quarter of a cell, (b) in a plane.

The space time notation widely adopted is indicated as

$$u|_{i,j,k}^n = u(i\Delta x, j\Delta y, k\Delta z, n\Delta t) \quad (2.1.3)$$

where Δx , Δy and Δz are the spatial steps in the x -, y - and z -directions respectively, Δt is the time step. i , j , k and n are the associated spatial and time indices.

Using the second order accurate central difference approximation for space and time derivatives in conjunction with the staggered Yee's cell, the explicit finite-difference approximation of the six scalar Maxwell's equations in (2.1.2) can be obtained, which form the update equations for electric fields and magnetic fields.

$$E_x|_{i+\frac{1}{2},j,k}^{n+1} = E_x|_{i+\frac{1}{2},j,k}^n + \frac{\Delta t}{\epsilon} \left(\frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta y} - \frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta z} \right) \quad (2.1.4a)$$

$$E_y|_{i,j+\frac{1}{2},k}^{n+1} = E_y|_{i,j+\frac{1}{2},k}^n + \frac{\Delta t}{\epsilon} \left(\frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta z} - \frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta x} \right) \quad (2.1.4b)$$

$$E_z|_{i,j,k+\frac{1}{2}}^{n+1} = E_z|_{i,j,k+\frac{1}{2}}^n$$

$$+ \frac{\Delta t}{\epsilon} \left(\frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} - \frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta y} \right) \quad (2.1.4c)$$

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{3}{2}} = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta t}{\mu} \left(\frac{E_y|_{i,j+\frac{1}{2},k+1}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1}}{\Delta z} - \frac{E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1}}{\Delta y} \right) \quad (2.1.4d)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{3}{2}} = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta t}{\mu} \left(\frac{E_z|_{i+1,j,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1}}{\Delta x} - \frac{E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1}}{\Delta z} \right) \quad (2.1.4e)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{3}{2}} = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta t}{\mu} \left(\frac{E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1}}{\Delta y} - \frac{E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1}}{\Delta x} \right) \quad (2.1.4f)$$

Note that apart from the staggered Yee's cell, the electric field and magnetic field components are evaluated at half time step away [2]. Therefore any electric field or magnetic field components at the present time step can be calculated from the field components at the previous time steps, which make the Yee's FDTD an explicit algorithm. It is well known that the Yee's explicit FDTD method has second order temporal and spatial accuracy with third order error terms of $O(\Delta t^3)$ and $O(\Delta \xi^3)$, where $\xi = x, y, z$.

Due to its explicit nature, the Yee's FDTD method is subjected to the CFL condition [3] given by

$$\Delta t \leq \sqrt{\frac{\mu\epsilon}{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}} \quad (2.1.5)$$

which imposes a maximum constraint on its time step size. It can be seen that the maximum allowed time step is restricted by the smallest spatial step used. In

some structures where there are combinations of fine and coarse grids, the time step is always restricted to the size of the fine grids, which degrades the efficiency significantly.

2.1.2 ADI-FDTD Method

To remove the CFL condition of the Yee's explicit FDTD method, unconditionally stable implicit algorithms have been introduced. The Crank-Nicolson based implicit FDTD (CN-FDTD) method [4–7] has been introduced to overcome the CFL condition. However, due to its implementation complexity, solving the updating equations are rather tedious and computationally intensive.

Another implicit method using the ADI-FDTD algorithm [8–11] has been developed. It is an unconditionally stable method which removes the CFL condition entirely. Since the time step is no longer restricted by the CFL constraint, the simulation time can be accelerated. Again, assuming a source-free, lossless, isotropic and non-dispersive medium, the 3-D ADI-FDTD method can be written in a compact matrix form as

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^{n+\frac{1}{2}} = \left(\mathbf{I}_6 + \frac{\Delta t}{2}\mathbf{B}_E\right)\mathbf{u}^n \quad (2.1.6a)$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{B}_E\right)\mathbf{u}^{n+1} = \left(\mathbf{I}_6 + \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^{n+\frac{1}{2}}. \quad (2.1.6b)$$

where

$$\mathbf{u} = [E_x, E_y, E_z, H_x, H_y, H_z]^T \quad (2.1.7)$$

$$\mathbf{A}_E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{\epsilon}\partial_y \\ 0 & 0 & 0 & \frac{1}{\epsilon}\partial_z & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\epsilon}\partial_x & 0 \\ 0 & \frac{1}{\mu}\partial_z & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\mu}\partial_x & 0 & 0 & 0 \\ \frac{1}{\mu}\partial_y & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.1.8)$$

$$\mathbf{B}_E = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{-1}{\epsilon}\partial_z & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-1}{\epsilon}\partial_x \\ 0 & 0 & 0 & \frac{-1}{\epsilon}\partial_y & 0 & 0 \\ 0 & 0 & \frac{-1}{\mu}\partial_y & 0 & 0 & 0 \\ \frac{-1}{\mu}\partial_z & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{\mu}\partial_x & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.1.9)$$

$\mathbf{I}_6$ is an 6×6 identity matrix. ∂_x , ∂_y and ∂_z are the spatial differential operators in the x -, y - and z -directions, respectively. Note that the above compact matrix equations are accompanied by the usage of the Yee's FDTD grid and central difference on the spatial derivatives. To assert the accuracy and error analysis for the ADI-FDTD method, the Maxwell's equations are cast into a compact matrix form as

$$\frac{\partial}{\partial t} \mathbf{u} = (\mathbf{A}_E + \mathbf{B}_E) \cdot \mathbf{u} \quad (2.1.10)$$

Assuming initial time of zero, the solution known to exist for such first order differential matrix system is expressed as

$$\mathbf{u}(t) = e^{(\mathbf{A}_E + \mathbf{B}_E)t} \cdot \mathbf{u}(0) \quad (2.1.11)$$

$\mathbf{A}_E$ and $\mathbf{B}_E$ involve space derivative operators and therefore the matrix exponential will contain exponential terms of space derivatives, which are almost impossible to implement. If second order Pade approximation is used for the matrix exponential,

it will result in

$$\mathbf{u}(t) = \left[\mathbf{I}_6 + \frac{1}{2} (\mathbf{A}_E + \mathbf{B}_E) t \right] \cdot \left[\mathbf{I}_6 - \frac{1}{2} (\mathbf{A}_E + \mathbf{B}_E) t \right]^{-1} \cdot \mathbf{u}(0) \quad (2.1.12)$$

The fields after one time step will be given as

$$\mathbf{u}(\Delta t) = \left[\mathbf{I}_6 + \frac{\Delta t}{2} (\mathbf{A}_E + \mathbf{B}_E) \right] \cdot \left[\mathbf{I}_6 - \frac{\Delta t}{2} (\mathbf{A}_E + \mathbf{B}_E) \right]^{-1} \cdot \mathbf{u}(0) \quad (2.1.13)$$

Using the following matrix properties

$$(\mathbf{I}_6 - \mathbf{A}_E)^{-1} \cdot (\mathbf{I}_6 + \mathbf{A}_E) = (\mathbf{I}_6 + \mathbf{A}_E) \cdot (\mathbf{I}_6 - \mathbf{A}_E)^{-1} \quad (2.1.14)$$

the update equations for the fields can be written as

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2} (\mathbf{A}_E + \mathbf{B}_E) \right) \cdot \mathbf{u}^{n+1} = \left(\mathbf{I}_6 + \frac{\Delta t}{2} (\mathbf{A}_E + \mathbf{B}_E) \right) \cdot \mathbf{u}^n \quad (2.1.15)$$

Equation (2.1.15) is known as the CN scheme and is of second order accurate in time due to the second order Pade approximation used initially at (2.1.11). Owing to the fact that the matrix inverse on the LHS of (2.1.15) is irreducible, splitting can be further performed, which forms the ADI-FDTD method. Prior to that, additional terms are added to both sides of (2.1.15) to yield

$$\begin{aligned} \left(\mathbf{I}_6 - \frac{\Delta t}{2} (\mathbf{A}_E + \mathbf{B}_E) + \left(\frac{\Delta t}{2} \right)^2 \mathbf{A}_E \cdot \mathbf{B}_E \right) \cdot \mathbf{u}^{n+1} = \\ \left(\mathbf{I}_6 + \frac{\Delta t}{2} (\mathbf{A}_E + \mathbf{B}_E) + \left(\frac{\Delta t}{2} \right)^2 \mathbf{A}_E \cdot \mathbf{B}_E \right) \cdot \mathbf{u}^n \end{aligned} \quad (2.1.16)$$

It can be shown that (2.1.16) still preserve the second order temporal accuracy because the additional terms will only introduce error at order three, $O(\Delta t^3)$. Factorizing, one will obtain

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{A}_E \right) \cdot \left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{B}_E \right) \cdot \mathbf{u}^{n+1} = \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{A}_E \right) \cdot \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{B}_E \right) \cdot \mathbf{u}^n \quad (2.1.17)$$

Multiplying both sides with $\left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{A}_E \right)^{-1}$, the update equations of the ADI-

FDTD method can be written in two procedures as in (2.1.6). Therefore the second order temporal accuracy of the existing ADI-FDTD is ascertained by observing the presence of third order error terms $O(\Delta t^3)$ in the overall updating procedure. Using the central difference approximation for space derivatives, the ADI-FDTD method has second order spatial accuracy with third order error terms $O(\Delta \xi^3)$. It is also noted that swapping the roles of $\mathbf{A}_E$ and $\mathbf{B}_E$ will not alter the overall temporal accuracy.

Expanding (2.1.6), upon some manipulations and arrangements, the full update equations of the ADI-FDTD method are given as

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta y^2}E_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta y^2}\right)E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} \\
& = E_x|_{i+\frac{1}{2},j,k}^n - \frac{\Delta t^2}{4\epsilon\mu\Delta y\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^n - E_y|_{i,j+\frac{1}{2},k}^n - E_y|_{i+1,j-\frac{1}{2},k}^n + E_y|_{i,j-\frac{1}{2},k}^n\right) \\
& \quad + \frac{\Delta t}{2\epsilon\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) - \frac{\Delta t}{2\epsilon\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^n\right) \tag{2.1.18a}
\end{aligned}$$

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta z^2}E_y|_{i,j+\frac{1}{2},k-1}^{n+\frac{1}{2}} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta z^2}\right)E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta z^2}E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} \\
& = E_y|_{i,j+\frac{1}{2},k}^n - \frac{\Delta t^2}{4\epsilon\mu\Delta z\Delta y} \left(E_z|_{i,j+1,k+\frac{1}{2}}^n - E_z|_{i,j,k+\frac{1}{2}}^n - E_z|_{i,j+1,k-\frac{1}{2}}^n + E_z|_{i,j,k-\frac{1}{2}}^n\right) \\
& \quad + \frac{\Delta t}{2\epsilon\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n\right) - \frac{\Delta t}{2\epsilon\Delta x} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^n\right) \tag{2.1.18b}
\end{aligned}$$

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta x^2}E_z|_{i-1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta x^2}\right)E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta x^2}E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\
& = E_z|_{i,j,k+\frac{1}{2}}^n - \frac{\Delta t^2}{4\epsilon\mu\Delta x\Delta z} \left(E_x|_{i+\frac{1}{2},j,k+1}^n - E_x|_{i+\frac{1}{2},j,k}^n - E_x|_{i-\frac{1}{2},j,k+1}^n + E_x|_{i-\frac{1}{2},j,k}^n\right) \\
& \quad + \frac{\Delta t}{2\epsilon\Delta x} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n\right) - \frac{\Delta t}{2\epsilon\Delta y} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^n\right) \tag{2.1.18c}
\end{aligned}$$

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n$$

$$+ \frac{\Delta t}{2\mu} \left(\frac{E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta z} - \frac{E_z|_{i,j+1,k+\frac{1}{2}}^n - E_z|_{i,j,k+\frac{1}{2}}^n}{\Delta y} \right) \quad (2.1.18d)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu} \left(\frac{E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} - \frac{E_x|_{i+\frac{1}{2},j,k+1}^n - E_x|_{i+\frac{1}{2},j,k}^n}{\Delta z} \right) \quad (2.1.18e)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\Delta t}{2\mu} \left(\frac{E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}}{\Delta y} - \frac{E_y|_{i+1,j+\frac{1}{2},k}^n - E_y|_{i,j+\frac{1}{2},k}^n}{\Delta x} \right) \quad (2.1.18f)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} & - \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} E_x|_{i+\frac{1}{2},j,k-1}^{n+1} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta z^2} \right) E_x|_{i+\frac{1}{2},j,k}^{n+1} - \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} E_x|_{i+\frac{1}{2},j,k+1}^{n+1} \\ & = E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta z\Delta x} \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i+1,j,k-\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \\ & + \frac{\Delta t}{2\epsilon\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) - \frac{\Delta t}{2\epsilon\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (2.1.19a)$$

$$\begin{aligned} & - \frac{\Delta t^2}{4\epsilon\mu\Delta x^2} E_y|_{i-1,j+\frac{1}{2},k}^{n+1} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta x^2} \right) E_y|_{i,j+\frac{1}{2},k}^{n+1} - \frac{\Delta t^2}{4\epsilon\mu\Delta x^2} E_y|_{i+1,j+\frac{1}{2},k}^{n+1} \\ & = E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta x\Delta y} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - E_x|_{i-\frac{1}{2},j+1,k}^{n+\frac{1}{2}} + E_x|_{i-\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \\ & + \frac{\Delta t}{2\epsilon\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}} \right) - \frac{\Delta t}{2\epsilon\Delta x} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \end{aligned} \quad (2.1.19b)$$

$$\begin{aligned} & - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} E_z|_{i,j-1,k+\frac{1}{2}}^{n+1} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta y^2} \right) E_z|_{i,j,k+\frac{1}{2}}^{n+1} - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} \\ & = E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta y\Delta z} \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - E_y|_{i,j-\frac{1}{2},k+1}^{n+\frac{1}{2}} + E_y|_{i,j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \\ & + \frac{\Delta t}{2\epsilon\Delta x} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) - \frac{\Delta t}{2\epsilon\Delta y} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (2.1.19c)$$

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}$$

$$+ \frac{\Delta t}{2\mu} \left(\frac{E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}}}{\Delta z} - \frac{E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1}}{\Delta y} \right) \quad (2.1.19d)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta t}{2\mu} \left(\frac{E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} - \frac{E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1}}{\Delta z} \right) \quad (2.1.19e)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta t}{2\mu} \left(\frac{E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}}{\Delta y} - \frac{E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1}}{\Delta x} \right) \quad (2.1.19f)$$

This method has a tridiagonal system of equations which can be solved efficiently through Thomas algorithm. Note that the electric field components are solved implicitly while the magnetic field components are then solved explicitly. Unlike the Yee's explicit FDTD method, the ADI-FDTD method has the electric field and magnetic field components evaluated at the same time step.

2.1.3 LOD-FDTD Method

The LOD-FDTD method, or split-step (SS) scheme [12–16], is another method with unconditional stability feature. This scheme, denoted as LOD (or SS1) here calls for the splitting formulae in the form of

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{A}_E \right) \mathbf{u}^{n+\frac{1}{2}} = \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{A}_E \right) \mathbf{u}^n \quad (2.1.20a)$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{B}_E \right) \mathbf{u}^{n+1} = \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{B}_E \right) \mathbf{u}^{n+\frac{1}{2}}. \quad (2.1.20b)$$

The above scheme has a greater efficiency compared to the ADI-FDTD. Similar to the ADI-FDTD method, to assert the accuracy and error analysis of the LOD-FDTD

method, we rewrite (2.1.11)

$$\mathbf{u}(t) = e^{\mathbf{B}_E t} \cdot e^{\mathbf{A}_E t} \cdot \mathbf{u}(0) \quad (2.1.21)$$

Applying second order Pade approximation to (2.1.21), the fields after one time step Δt will be given as

$$\mathbf{u}(\Delta t) = \left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{A}_E \right)^{-1} \cdot \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{A}_E \right) \cdot \left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{B}_E \right)^{-1} \cdot \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{B}_E \right) \cdot \mathbf{u}(0) \quad (2.1.22)$$

The update equations for the LOD-FDTD will follow suit in two procedures as in (2.1.20). It is noted that the approximation used in (2.1.21) is only of first order in temporal accuracy due to the non-commutative nature of $\mathbf{A}_E$ and $\mathbf{B}_E$. Therefore, despite the fact that second order Pade approximation is used, the LOD-FDTD scheme will only have first order temporal accuracy due to the presence of second order error terms $O(\Delta t^2)$ as observed in the overall updating procedure. Using the central difference approximation for space derivatives, the LOD-FDTD method has second order spatial accuracy with third order error terms $O(\Delta \xi^3)$.

By expanding (2.1.20), upon some manipulation and arrangements, the full update equations of the LOD-FDTD method are shown below:

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} & -\frac{\Delta t^2}{4\epsilon\mu\Delta y^2} E_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta y^2}\right) E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} \\ & = \left(1 - \frac{\Delta t^2}{2\epsilon\mu\Delta y^2}\right) E_x|_{i+\frac{1}{2},j,k}^n + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} \left(E_x|_{i+\frac{1}{2},j-1,k}^n + E_x|_{i+\frac{1}{2},j+1,k}^n\right) \\ & \quad + \frac{\Delta t}{\epsilon\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) \end{aligned} \quad (2.1.23a)$$

$$\begin{aligned} & -\frac{\Delta t^2}{4\epsilon\mu\Delta z^2} E_y|_{i,j+\frac{1}{2},k-1}^{n+\frac{1}{2}} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta z^2}\right) E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} \\ & = \left(1 - \frac{\Delta t^2}{2\epsilon\mu\Delta z^2}\right) E_y|_{i,j+\frac{1}{2},k}^n + \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} \left(E_y|_{i,j+\frac{1}{2},k-1}^n + E_y|_{i,j+\frac{1}{2},k+1}^n\right) \\ & \quad + \frac{\Delta t}{\epsilon\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n\right) \end{aligned} \quad (2.1.23b)$$

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta x^2}E_z|_{i-1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta x^2}\right)E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta x^2}E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\
& = \left(1 - \frac{\Delta t^2}{2\epsilon\mu\Delta x^2}\right)E_z|_{i,j,k+\frac{1}{2}}^n + \frac{\Delta t^2}{4\epsilon\mu\Delta x^2}\left(E_z|_{i-1,j,k+\frac{1}{2}}^n + E_z|_{i+1,j,k+\frac{1}{2}}^n\right) \\
& \quad + \frac{\Delta t}{\epsilon\Delta x}\left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n\right) \tag{2.1.23c}
\end{aligned}$$

$$\begin{aligned}
H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} & = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n \\
& \quad + \frac{\Delta t}{2\mu\Delta z}\left(E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + E_y|_{i,j+\frac{1}{2},k+1}^n - E_y|_{i,j+\frac{1}{2},k}^n\right) \tag{2.1.23d}
\end{aligned}$$

$$\begin{aligned}
H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} & = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n \\
& \quad + \frac{\Delta t}{2\mu\Delta x}\left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i+1,j,k+\frac{1}{2}}^n - E_z|_{i,j,k+\frac{1}{2}}^n\right) \tag{2.1.23e}
\end{aligned}$$

$$\begin{aligned}
H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} & = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n \\
& \quad + \frac{\Delta t}{2\mu\Delta y}\left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j+1,k}^n - E_x|_{i+\frac{1}{2},j,k}^n\right) \tag{2.1.23f}
\end{aligned}$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta z^2}E_x|_{i+\frac{1}{2},j,k-1}^{n+1} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta z^2}\right)E_x|_{i+\frac{1}{2},j,k}^{n+1} - \frac{\Delta t^2}{4\epsilon\mu\Delta z^2}E_x|_{i+\frac{1}{2},j,k+1}^{n+1} \\
& = \left(1 - \frac{\Delta t^2}{2\epsilon\mu\Delta z^2}\right)E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\Delta t^2}{4\epsilon\mu\Delta z^2}\left(E_x|_{i+\frac{1}{2},j,k-1}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,k+1}^{n+\frac{1}{2}}\right) \\
& \quad - \frac{\Delta t}{\epsilon\Delta z}\left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{2.1.24a}
\end{aligned}$$

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta x^2}E_y|_{i-1,j+\frac{1}{2},k}^{n+1} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta x^2}\right)E_y|_{i,j+\frac{1}{2},k}^{n+1} - \frac{\Delta t^2}{4\epsilon\mu\Delta x^2}E_y|_{i+1,j+\frac{1}{2},k}^{n+1} \\
& = \left(1 - \frac{\Delta t^2}{2\epsilon\mu\Delta x^2}\right)E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta t^2}{4\epsilon\mu\Delta x^2}\left(E_y|_{i-1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + E_y|_{i+1,j+\frac{1}{2},k}^{n+\frac{1}{2}}\right) \\
& \quad - \frac{\Delta t}{\epsilon\Delta x}\left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}\right) \tag{2.1.24b}
\end{aligned}$$

$$\begin{aligned}
& -\frac{\Delta t^2}{4\epsilon\mu\Delta y^2}E_z|_{i,j-1,k+\frac{1}{2}}^{n+1} + \left(1 + \frac{\Delta t^2}{2\epsilon\mu\Delta y^2}\right)E_z|_{i,j,k+\frac{1}{2}}^{n+1} - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} \\
& = \left(1 - \frac{\Delta t^2}{2\epsilon\mu\Delta y^2}\right)E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\left(E_z|_{i,j-1,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,j+1,k+\frac{1}{2}}^{n+\frac{1}{2}}\right) \\
& \quad - \frac{\Delta t}{\epsilon\Delta y}\left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{2.1.24c}
\end{aligned}$$

$$\begin{aligned}
H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} & = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \\
& \quad - \frac{\Delta t}{2\mu\Delta y}\left(E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1} + E_z|_{i,j+1,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{2.1.24d}
\end{aligned}$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta z} \left(E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} + E_x|_{i+\frac{1}{2},j,k+1}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (2.1.24e)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} + E_y|_{i+1,j+\frac{1}{2},k}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (2.1.24f)$$

Similar to the ADI-FDTD method, the electric field components which have a tridiagonal system of equations are solved implicitly through Thomas algorithm while the magnetic field components are then solved explicitly. Comparing the update equations (2.1.18)-(2.1.19) and (2.1.23)-(2.1.24) for both methods, it can be seen that the number of arithmetic operations is reduced, thus resulting in a greater efficiency for the LOD-FDTD method.

2.1.4 Graphics Processing Units

Despite having unconditionally stable ADI- and/or LOD-FDTD methods with its time step unrestricted by the CFL condition, these methods come with some shortcomings such as increased complexity due to the substantial amount of arithmetic operations and field variables involved in the RHS of its update equations. As CPU consist of a few cores optimized for sequential serial processing, GPU computing has been introduced to accelerate the computation time. Programmable GPUs with highly parallel processors have led to the interest in using GPUs for general purpose programming [17–20]. Such highly parallel processing feature of the GPU motivates the implementation of ADI- and/or LOD-FDTD methods.

CUDA is a parallel computing architecture developed by NVIDIA [17–20], which is accessible to software developers through industry standard programming languages. Figure 2.3 gives an overview of the device architecture. A GPU with CUDA capabilities has a set of build-in streaming multiprocessors (SMs). Each SM consists of

Figure 2.3: Device architecture.

a shared memory (which can be used to share data between the threads within a thread block), a set of 32-bit registers and read-only caches. Each multiprocessor contains 8 streaming processors, and every streaming processor supports parallel executing model of the single-instruction multiple-data (SIMD). The Random Access Memory (RAM) located on the GPU card serves as the global memory used to store massive data.

A CUDA program is executed on both the host (CPU) and device (GPU). The set of instructions that exhibit rich amount of data parallelism are implemented in the device code, whereas the set of instructions with little or no data parallelism are implemented in the host code.

Figure 2.4: Execution of a Typical CUDA Program.

The execution of a typical CUDA program (illustrated in Figure 2.4) starts with host execution [20]. When a kernel function is invoked, the execution is moved to a device, where a large number of threads are generated to take advantage of abundant data parallelism. All the threads that are generated by a kernel are organized as a grid of thread blocks and each block consists of a maximum of 512 threads. The grid of thread blocks is then executed on the GPU by assigning blocks for execution on the SMs. Upon receiving a block, the multiprocessor divides it into groups of 32 threads named warps. Then, warps are executed one common instruction at a time. When the execution of a kernel is completed, the corresponding grid terminates and the execution returns to the host until another kernel is invoked.

2.1.5 Absorbing Boundary Conditions

Because computational storage is finite, the FDTD problem space size is finite and needs to be truncated by special boundary conditions. The types of special boundary conditions that simulate electromagnetic waves propagating continuously beyond the computational space are called ABCs. However, the imperfect truncation of the

problem space will create numerical reflections, which will corrupt the computational results in the problem space after certain amount of simulation time. Generally, there are two ways to construct ABCs. One method is the usage of the outgoing wave equation. Engquist and Majda [21] are first to use the outgoing wave equation to construct ABCs. Since then, numerous ABCs [22–32] are derived through the outgoing wave equation. Among them, Mur ABC [33–38] is the most widely used for its simplicity and reasonable absorption (in terms of reflection coefficient).

Mur ABC

Mur introduced a simple and successful finite-difference scheme for the ABCs based on a partial differential equation that permits wave propagation only in certain directions called a one-way wave equation. Consider a general 3-D wave equation in the Cartesian coordinate system where electric field components E_ξ are the tangential fields on the boundary,

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} \right) E_\xi = 0 \quad (2.1.25)$$

where v is the wave phase velocity. For a plane wave that propagates from the $z > 0$ region and impinges on the $z = 0$ boundary, it is found that there will be no reflection from the boundary if the following wave equation is satisfied:

$$\left(\frac{\partial}{\partial z} + \frac{1}{v} \frac{\partial}{\partial t} [1 - (v\tilde{s}_x)^2 - (v\tilde{s}_y)^2]^{\frac{1}{2}} \right) E_\xi|_{z=0} = 0 \quad (2.1.26)$$

where $\tilde{\xi} = x, y$ and $\tilde{s}_x^2 + \tilde{s}_y^2 + \tilde{s}_z^2 = v^{-2}$. However, due to the fact that the incident angle of the wave reaching the boundary is unknown, an assumption for the term $[1 - (v\tilde{s}_x)^2 - (v\tilde{s}_y)^2]^{\frac{1}{2}}$ in (2.1.26) is required. Using Taylor series expansions, Mur proposed a first order ABC by assuming

$$[1 - (v\tilde{s}_x)^2 - (v\tilde{s}_y)^2]^{\frac{1}{2}} = 1 + 0 \left((v\tilde{s}_x)^2 - (v\tilde{s}_y)^2 \right) \quad (2.1.27)$$

The corresponding one-way wave equation for the Mur first order ABC is

$$\left(\frac{\partial}{\partial z} - \frac{1}{v} \frac{\partial}{\partial t} \right) E_{\xi}|_{z=0} = 0 \quad (2.1.28)$$

Through manipulations of the time and space derivatives, we can incorporate the Mur ABC into the Yee's explicit FDTD method (illustrated here for E_x variable at the boundary $z = 0$) as

$$\frac{\partial E_x|_{i+\frac{1}{2},j,\frac{1}{2}}^{n+\frac{1}{2}}}{\partial t} = v \frac{\partial E_x|_{i+\frac{1}{2},j,\frac{1}{2}}^{n+\frac{1}{2}}}{\partial z} \quad (2.1.29)$$

The updating equation for E_x variable at the boundary $z = 0$ is discretized as follows:

$$E_x|_{i+\frac{1}{2},j,0}^{n+1} = E_x|_{i+\frac{1}{2},j,1}^n + \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z} \left(E_x|_{i+\frac{1}{2},j,1}^{n+1} - E_x|_{i+\frac{1}{2},j,0}^n \right) \quad (2.1.30)$$

The implementation of the Mur ABC is relatively convenient with low memory consumption and reasonable good reflection coefficient. The complete update equations of the Mur first order ABC for Yee's explicit FDTD method are in Appendix A.

Perfectly Matched Layers

Another method to construct ABC is the employment of nonphysical absorbing media, such as the PML [39–51] which can absorb wide-angle scattered wave, but requires extra computer memory and is highly complex.

The PML introduced by Berenger [39–41] is a major breakthrough for ABCs in comparison with the other techniques adopted in the past. PML is a finite-thickness special medium surrounding the computational space based on fictitious constitutive parameters to create a wave-impedance matching condition, which is independent of the angles and frequencies of the wave incident on this boundary.

Figure 2.5: Structure of a 2-D TE, FDTD grid employing the Berenger PML ABC in [39].

In the Berenger's original split-field formulation [39], the 2-D transverse electric (TE) FDTD which consists of E_x , E_y , H_{zx} and H_{zy} field components satisfy the following relations:

$$\epsilon \frac{\partial E_x}{\partial t} + \sigma_y E_x = \frac{\partial (H_{zx} + H_{zy})}{\partial y} \quad (2.1.31a)$$

$$\epsilon \frac{\partial E_y}{\partial t} + \sigma_x E_y = -\frac{\partial (H_{zx} + H_{zy})}{\partial x} \quad (2.1.31b)$$

$$\mu \frac{\partial H_{zx}}{\partial t} + \sigma_x^m H_{zx} = -\frac{\partial E_y}{\partial x} \quad (2.1.31c)$$

$$\mu \frac{\partial H_{zy}}{\partial t} + \sigma_y^m H_{zy} = \frac{\partial E_x}{\partial y}. \quad (2.1.31d)$$

Here, H_z is assumed to be split into two additive subcomponents.

$$H_z = H_{zx} + H_{zy} \quad (2.1.32)$$

Further, the parameters σ_x and σ_y denote electric conductivities and the parameters σ_x^m and σ_y^m denote magnetic conductivities in the PML region.

Figure 2.5 presents the structure of a 2-D TE, FDTD grid employing the Berenger PML ABC in [39]. We see that Berenger's formulation represents a generalization of normally modeled physical media. If $\sigma_x = \sigma_y = 0$ and $\sigma_x^m = \sigma_y^m = 0$, (2.1.31) reduces to Maxwell's equations in a lossless medium. If $\sigma_x = \sigma_y = \sigma$ (where σ is the electric conductivity) and $\sigma_x^m = \sigma_y^m = 0$, (2.1.31) describes an electrically conductive medium. If $\epsilon_1 = \epsilon_2$, $\mu_1 = \mu_2$, $\sigma_x = \sigma_y = \sigma$, $\sigma_x^m = \sigma_y^m = \sigma^m$ (where σ^m is the magnetic conductivity) and $\frac{\sigma^m}{\mu_1} = \frac{\sigma}{\epsilon_1}$ are satisfied, then (2.1.31) describes an absorbing medium that is impedance-matched to the normally incident plane waves.

The split-field PML absorbing media has proven to be the most robust and efficient technique for the termination of FDTD lattices. However, it is ineffective at absorbing evanescent waves and suffers from late-time reflections when terminating highly elongated lattices or when simulating fields with very long time-signatures.

In 2000, Roden and Gedney presented an efficient implementation of Berenger's PML called the CPML [46]. With the introduction of the CFS into the CPML, evanescent waves and signals of long time-signature can be absorbed effectively. This method consists of mapping Maxwell's equations into a complex stretched coordinate space as

$$j\omega\epsilon\mathbf{E} = \tilde{\nabla} \times \mathbf{H} \quad (2.1.33a)$$

$$-j\omega\mu\mathbf{H} = \tilde{\nabla} \times \mathbf{E} \quad (2.1.33b)$$

where

$$\tilde{\nabla} = \hat{\mathbf{x}} \frac{1}{s_x} \frac{\partial}{\partial x} + \hat{\mathbf{y}} \frac{1}{s_y} \frac{\partial}{\partial y} + \hat{\mathbf{z}} \frac{1}{s_z} \frac{\partial}{\partial z}, \quad (2.1.34)$$

$j = \sqrt{-1}$, $\omega = 2\pi f$, f represents frequency and the stretched coordinate metric is given as

$$s_\xi = \hat{\kappa}_\xi + \frac{\sigma_\xi}{\hat{\alpha}_\xi + j\omega\epsilon} \quad (2.1.35)$$

where $\hat{\alpha}_\xi$ is positive real added to the stretching factor [44].

Within the PML, the constitutive parameters are scaled using polynomial scaling

$$\sigma_\xi(\xi) = \frac{\sigma_{\xi_{max}} |\xi - \xi_0|^m}{\delta^m} \quad (2.1.36)$$

$$\hat{\kappa}_\xi(\xi) = 1 + (\hat{\kappa}_{\xi_{max}} - 1) \frac{|\xi - \xi_0|^m}{\delta^m} \quad (2.1.37)$$

where δ is the thickness of the PML absorber, ξ_0 is the interface to free space and m is the order of the polynomial.

A choice for $\sigma_{\xi_{max}}$ that will minimize reflection is expressed as [39]

$$\sigma_{\xi_{max}} = \sigma_{opt} \approx \frac{m+1}{150\pi\Delta\xi}. \quad (2.1.38)$$

Expanding (2.1.33), upon some manipulations and arrangements, the update equations of the Yee's explicit FDTD method with CFS-CPML are as follows (other field equations can be written down by permuting the indices):

$$\begin{aligned} E_x|_{i+\frac{1}{2},j,k}^{n+1} = & E_x|_{i+\frac{1}{2},j,k}^n \\ & + \frac{\Delta t}{\epsilon} \left(\frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}}{\hat{\kappa}_{y_j}\Delta y} - \frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}}{\hat{\kappa}_{z_k}\Delta z} \right. \\ & \left. + \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \end{aligned} \quad (2.1.39a)$$

$$\begin{aligned} H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{3}{2}} = & H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \\ & + \frac{\Delta t}{\mu} \left(\frac{E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1}}{\hat{\kappa}_{y_{j+\frac{1}{2}}}\Delta y} - \frac{E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1}}{\hat{\kappa}_{x_{i+\frac{1}{2}}}\Delta x} \right. \\ & \left. + \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} \right) \end{aligned} \quad (2.1.39b)$$

Note that $\psi_{e_{xy}}$, $\psi_{e_{xz}}$, $\psi_{h_{zx}}$ and $\psi_{h_{zy}}$ are discrete variables with nonzero values only

in their corresponding PML regions. The update equations of these variables are

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \widehat{c}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} + \frac{\widehat{b}_{y_j}}{\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (2.1.40a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \widehat{c}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} + \frac{\widehat{b}_{z_k}}{\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (2.1.40b)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \widehat{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\widehat{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (2.1.40c)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \widehat{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\widehat{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (2.1.40d)$$

where

$$\widehat{c}_\xi = e^{-\left(\frac{\sigma_\xi}{\kappa_\xi} + \widehat{\alpha}_\xi\right) \frac{\Delta t}{\epsilon}}, \quad (2.1.41)$$

$$\widehat{b}_\xi = \frac{\sigma_\xi}{\sigma_\xi \widehat{\kappa}_\xi + \widehat{\kappa}_\xi^2 \widehat{\alpha}_\xi} \left(\widehat{c}_\xi - 1 \right). \quad (2.1.42)$$

The full update equations of the Yee's explicit FDTD method with CFS-CPML and their discrete variables are in Appendix B. Note that there are only two discrete variables required for each field point. The CFS-CPML is highly robust in the absorption of waves in inhomogeneous, lossy, dispersive, anisotropic, or nonlinear media. However, this comes with an increased complexity of the update equations and incurs high computer memory.

2.2 FDTD Methods for Heat Transfer

In the previous section, various FDTD methods have been used on the Maxwell's equations for modeling computational electromagnetics. Since the FDTD is a numerical analysis technique used to find approximate solutions to partial differential equations. Therefore, these FDTD methods can also be used in other branches of physics. In computational thermodynamics, the heat transfer equation is a parabolic partial differential equation used to describe the distribution of heat in a given region over time. Based on the FDTD technique used for solving Maxwell's equation

in electromagnetics, it can be extended to solve the heat transfer equation which will be discussed below.

The temperature of a system is governed by the following parabolic partial differential equation of the heat transfer equation [52, 53]:

$$\rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial t} = \nabla \cdot [\kappa(\vec{r})\nabla T(\vec{r}, t)] + g(\vec{r}, t) \quad (2.2.1)$$

subjected to boundary condition [54]

$$\kappa(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial n_\xi} + h_\xi T(\vec{r}, t) = f(\vec{r}, t) \quad (2.2.2)$$

where T is the time-dependent temperature at any point, κ is the thermal conductivity (W/m·K), ρ is the density of the material (kg/m³), C_p is the specific heat capacity (J/kg·K) and g is the heat energy generation rate (W/m³). $f(\vec{r}, t)$ is an arbitrary function on the boundary, $\partial/\partial n_\xi$ is the differentiation along the outward direction normal to the boundary and h_ξ is the equivalent heat transfer coefficient on the boundary. Note that for convection boundary condition, the function $f(\vec{r}, t)$ in (2.2.2) is $f(\vec{r}, t) = h_\xi T_\infty$, where T_∞ is a pre-specified temperature.

To simulate temperature distribution using finite-difference methods [55], discretization is performed in both space and time domains. For homogeneous material, the term $\nabla \cdot [\kappa(\vec{r})\nabla T(\vec{r}, t)]$ in (2.2.1) can be replaced by $\kappa(\vec{r})\nabla^2 T(\vec{r}, t)$. Then the heat transfer equation can be expanded into

$$\begin{aligned} \rho(x, y, z)C_p(x, y, z)\frac{\partial T(x, y, z, t)}{\partial t} = \kappa(x, y, z) & \left(\frac{\partial^2 T(x, y, z, t)}{\partial x^2} + \frac{\partial^2 T(x, y, z, t)}{\partial y^2} \right. \\ & \left. + \frac{\partial^2 T(x, y, z, t)}{\partial z^2} \right) + g(x, y, z, t). \end{aligned} \quad (2.2.3)$$

Next, we define a grid point in the solution region as

$$(i, j, k) \equiv (i\Delta x, j\Delta y, k\Delta z),$$

and any function of space and time as

$$T|_{i,j,k}^n \equiv T(i\Delta x, j\Delta y, k\Delta z, n\Delta t).$$

Taking finite-difference in time on the LHS of (2.2.3), we obtain

$$\frac{T|_{i,j,k}^{n+1} - T|_{i,j,k}^n}{\Delta t} = \frac{\kappa_{i,j,k}}{\rho_{i,j,k}C_{p_{i,j,k}}} \left(\frac{\partial^2 T|_{i,j,k}^\diamond}{\partial x^2} + \frac{\partial^2 T|_{i,j,k}^\diamond}{\partial y^2} + \frac{\partial^2 T|_{i,j,k}^\diamond}{\partial z^2} \right) + \frac{1}{\rho_{i,j,k}C_{p_{i,j,k}}} g|_{i,j,k}^n. \quad (2.2.4)$$

Note that on the RHS of (2.2.4), the time step is represented by $\diamond$ to cater for different time marching choices that can be made [53].

2.2.1 Explicit Method

Applying time step n for $\diamond$ on the RHS of (2.2.4), we obtain the explicit finite-difference method for thermal simulation as

$$\frac{T|_{i,j,k}^{n+1} - T|_{i,j,k}^n}{\Delta t} = \frac{\kappa_{i,j,k}}{\rho_{i,j,k}C_{p_{i,j,k}}} \left(\frac{\partial^2 T|_{i,j,k}^n}{\partial x^2} + \frac{\partial^2 T|_{i,j,k}^n}{\partial y^2} + \frac{\partial^2 T|_{i,j,k}^n}{\partial z^2} \right) + \frac{1}{\rho_{i,j,k}C_{p_{i,j,k}}} g|_{i,j,k}^n. \quad (2.2.5)$$

The solution is simple as there is only one unknown in each equation for every grid point. Furthermore, this method has second-order accuracy in space (error term $O(\Delta\xi^3)$) and first-order accuracy in time (error term $O(\Delta t^2)$). However, the explicit method has a stability constraint

$$\gamma = \vartheta_{i,j,k} \Delta t \left(\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2} \right) \leq \frac{1}{2} \quad (2.2.6)$$

where $\vartheta_{i,j,k} = \frac{\kappa_{i,j,k}}{\rho_{i,j,k}C_{p_{i,j,k}}}$. This restricts the time step size Δt for certain given spatial steps of Δx , Δy and Δz .

2.2.2 Implicit Methods

Simple Implicit Method

As can be seen in Section 2.2.1, the explicit finite-difference method has a stability constraint shown in (2.2.6). To overcome the stability constraint, the simple implicit method (backward Euler method) can be introduced by applying the time step $n + 1$ on the RHS for $\diamond$ of (2.2.4) to obtain

$$\frac{T|_{i,j,k}^{n+1} - T|_{i,j,k}^n}{\Delta t} = \frac{\kappa_{i,j,k}}{\rho_{i,j,k} C_{p_{i,j,k}}} \left(\frac{\partial^2 T|_{i,j,k}^{n+1}}{\partial x^2} + \frac{\partial^2 T|_{i,j,k}^{n+1}}{\partial y^2} + \frac{\partial^2 T|_{i,j,k}^{n+1}}{\partial z^2} \right) + \frac{1}{\rho_{i,j,k} C_{p_{i,j,k}}} g|_{i,j,k}^n. \quad (2.2.7)$$

This method has error terms $O(\Delta \xi^3)$ and $O(\Delta t^2)$, which has the same accuracy as the explicit method in (2.2.5). The solution requires seven unknowns to be solved at every grid point at time step $n + 1$.

Crank-Nicolson (CN) Method

Another unconditionally stable implicit method developed by Crank and Nicolson [56–59] can be implemented. By applying the time step $n + \frac{1}{2}$ for $\diamond$ on the RHS of (2.2.4) followed by time averaging, we obtain

$$\begin{aligned} \frac{T|_{i,j,k}^{n+1} - T|_{i,j,k}^n}{\Delta t} = & \frac{\kappa_{i,j,k}}{\rho_{i,j,k} C_{p_{i,j,k}}} \left(\frac{\partial^2 T|_{i,j,k}^{n+1} + \partial^2 T|_{i,j,k}^n}{2\partial x^2} + \frac{\partial^2 T|_{i,j,k}^{n+1} + \partial^2 T|_{i,j,k}^n}{2\partial y^2} \right. \\ & \left. + \frac{\partial^2 T|_{i,j,k}^{n+1} + \partial^2 T|_{i,j,k}^n}{2\partial z^2} \right) + \frac{1}{\rho_{i,j,k} C_{p_{i,j,k}}} g|_{i,j,k}^n. \end{aligned} \quad (2.2.8)$$

This will result in seven unknowns that are required to be solved in each equation at time step $n + 1$. Although there is an increased in complexity in the updating equation, this method has a higher accuracy of second-order in both time and space (error terms $O(\Delta \xi^3)$ and $O(\Delta t^3)$).

ADI Methods

The PR- and DG-ADI methods [60–64] involve solving (2.2.8) by splitting it into three procedures of different time steps (from n to $n + \frac{1}{3}$, $n + \frac{1}{3}$ to $n + \frac{2}{3}$ and $n + \frac{2}{3}$ to $n + 1$). This results in a tridiagonal matrix which can be solved efficiently by Thomas algorithm. However, in the 3-D case, only the DG-ADI method remains unconditionally stable while the PR-ADI method is only conditionally stable. Therefore, we will only present the DG-ADI method for the following section. To the best of the author's knowledge, the DG-ADI method may still have some stability issues in the inhomogeneous case, which will be discussed in Chapter 4.

2.2.3 Douglas-Gunn (DG) ADI Method

Douglas and Gunn developed an unconditionally stable ADI method [60] with an accuracy of second-order in both time and space (error terms $O(\Delta\xi^3)$ and $O(\Delta t^3)$). According to the DG-ADI method, (2.2.8) is solved in three procedures as

For first procedure from n to $n + \frac{1}{3}$:

$$T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} = \frac{\hat{r}_{i,j,k}}{2} \frac{\partial^2}{\partial x^2} \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) + \hat{r}_{i,j,k} \left(\frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) T_{\text{DG}}|_{i,j,k}^n + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \quad (2.2.9a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} = \frac{\hat{r}_{i,j,k}}{2} \frac{\partial^2}{\partial x^2} \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) + \frac{\hat{r}_{i,j,k}}{2} \frac{\partial^2}{\partial y^2} \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) + \hat{r}_{i,j,k} \frac{\partial^2}{\partial z^2} T_{\text{DG}}|_{i,j,k}^n + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \quad (2.2.9b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$T_{\text{DG}}|_{i,j,k}^{n+1} = \frac{\hat{r}_{i,j,k}}{2} \frac{\partial^2}{\partial x^2} \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) + \frac{\hat{r}_{i,j,k}}{2} \frac{\partial^2}{\partial y^2} \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j,k}^n \right)$$

$$+ \frac{\hat{r}_{i,j,k}}{2} \frac{\partial^2}{\partial z^2} (T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n) + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \quad (2.2.9c)$$

where

$$\hat{r}_{i,j,k} = \frac{\kappa_{i,j,k} \Delta t}{\rho_{i,j,k} C_{p_{i,j,k}}}, G|_{i,j,k}^n = \frac{\Delta t}{\rho_{i,j,k} C_{p_{i,j,k}}} g|_{i,j,k}^n.$$

By applying central approximation and upon some arithmetic manipulations, we get the following:

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} & -\frac{1}{2} \hat{r}_{x_{i,j,k}} T_{\text{DG}}|_{i-1,j,k}^{n+\frac{1}{3}} + (1 + \hat{r}_{x_{i,j,k}}) T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} - \frac{1}{2} \hat{r}_{x_{i,j,k}} T_{\text{DG}}|_{i+1,j,k}^{n+\frac{1}{3}} \\ & = \frac{1}{2} \hat{r}_{x_{i,j,k}} (T_{\text{DG}}|_{i-1,j,k}^n + T_{\text{DG}}|_{i+1,j,k}^n) + \hat{r}_{y_{i,j,k}} (T_{\text{DG}}|_{i,j-1,k}^n + T_{\text{DG}}|_{i,j+1,k}^n) + G|_{i,j,k}^n \\ & + \hat{r}_{z_{i,j,k}} (T_{\text{DG}}|_{i,j,k-1}^n + T_{\text{DG}}|_{i,j,k+1}^n) + (1 - \hat{r}_{x_{i,j,k}} - 2\hat{r}_{y_{i,j,k}} - 2\hat{r}_{z_{i,j,k}}) T_{\text{DG}}|_{i,j,k}^n \end{aligned} \quad (2.2.10a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} & -\frac{1}{2} \hat{r}_{y_{i,j,k}} T_{\text{DG}}|_{i,j-1,k}^{n+\frac{2}{3}} + (1 + \hat{r}_{y_{i,j,k}}) T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} - \frac{1}{2} \hat{r}_{y_{i,j,k}} T_{\text{DG}}|_{i,j+1,k}^{n+\frac{2}{3}} \\ & = \frac{1}{2} \hat{r}_{x_{i,j,k}} (T_{\text{DG}}|_{i-1,j,k}^{n+\frac{1}{3}} - 2T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i+1,j,k}^{n+\frac{1}{3}}) + G|_{i,j,k}^n \\ & + \frac{1}{2} \hat{r}_{x_{i,j,k}} (T_{\text{DG}}|_{i-1,j,k}^n + T_{\text{DG}}|_{i+1,j,k}^n) + \frac{1}{2} \hat{r}_{y_{i,j,k}} (T_{\text{DG}}|_{i,j-1,k}^n + T_{\text{DG}}|_{i,j+1,k}^n) \\ & + \hat{r}_{z_{i,j,k}} (T_{\text{DG}}|_{i,j,k-1}^n + T_{\text{DG}}|_{i,j,k+1}^n) + (1 - \hat{r}_{x_{i,j,k}} - \hat{r}_{y_{i,j,k}} - 2\hat{r}_{z_{i,j,k}}) T_{\text{DG}}|_{i,j,k}^n \end{aligned} \quad (2.2.10b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned} & -\frac{1}{2} \hat{r}_{z_{i,j,k}} T_{\text{DG}}|_{i,j,k-1}^{n+1} + (1 + \hat{r}_{z_{i,j,k}}) T_{\text{DG}}|_{i,j,k}^{n+1} - \frac{1}{2} \hat{r}_{z_{i,j,k}} T_{\text{DG}}|_{i,j,k+1}^{n+1} \\ & = \frac{1}{2} \hat{r}_{x_{i,j,k}} (T_{\text{DG}}|_{i-1,j,k}^{n+\frac{1}{3}} - 2T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i+1,j,k}^{n+\frac{1}{3}}) + G|_{i,j,k}^n \\ & + \frac{1}{2} \hat{r}_{y_{i,j,k}} (T_{\text{DG}}|_{i,j-1,k}^{n+\frac{2}{3}} - 2T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j+1,k}^{n+\frac{2}{3}}) \\ & + \frac{1}{2} \hat{r}_{x_{i,j,k}} (T_{\text{DG}}|_{i-1,j,k}^n + T_{\text{DG}}|_{i+1,j,k}^n) + \frac{1}{2} \hat{r}_{y_{i,j,k}} (T_{\text{DG}}|_{i,j-1,k}^n + T_{\text{DG}}|_{i,j+1,k}^n) \\ & + \frac{1}{2} \hat{r}_{z_{i,j,k}} (T_{\text{DG}}|_{i,j,k-1}^n + T_{\text{DG}}|_{i,j,k+1}^n) + (1 - \hat{r}_{x_{i,j,k}} - \hat{r}_{y_{i,j,k}} - \hat{r}_{z_{i,j,k}}) T_{\text{DG}}|_{i,j,k}^n \end{aligned} \quad (2.2.10c)$$

where $\widehat{r}_{\xi_{i,j,k}} = \frac{\widehat{r}_{i,j,k}}{\Delta \xi^2}$.

The 3-D DG-ADI method splits the heat transfer equation in (2.2.3) into three procedures, n to $n + \frac{1}{3}$, $n + \frac{1}{3}$ to $n + \frac{2}{3}$ and $n + \frac{2}{3}$ to $n + 1$. For the first procedure from n to $n + \frac{1}{3}$, the x -direction is solved implicitly using the tridiagonal system of equations, while the y - and z -directions are solved explicitly. Similarly, the second and third procedures solved the y - and z -directions implicitly while the remaining directions are solved explicitly.

As the traditional thermal simulation to solve the heat transfer equation is computationally intensive, it requires a large number of linear equations given by the equivalent thermal circuit. To significantly improve the simulation efficiency, exploiting the parallelism of simulation algorithms on multicore computing platform such as GPU [17–20] can be implemented. Since CPU consist of a few cores optimized for sequential serial processing, this in turn increases the CPU computation time. In order to accelerate the computation time, GPU computing has been introduced. The GPU consists of thousands of smaller, more efficient cores designed for handling multiple tasks simultaneously. GPU-accelerated computing is the use of a GPU together with a CPU to accelerate the FDTD methods.

2.3 FDTD Methods for Schrödinger Equation

The FDTD technique has been used in two different branches of physics: (1) Modeling computational electromagnetics using Maxwell's equations in Section 2.1, (2) Simulate and study computational thermodynamics using the heat transfer equation in Section 2.2. Another branch of physics to be considered here is quantum mechanics. In quantum mechanics, the Schrödinger equation is a partial differential equation that describes how the quantum state of some physical system changes with time. Using the FDTD technique for solving the heat transfer equation, this

technique can be extended to solve the Schrödinger equation.

The time-dependent Schrödinger equation is a parabolic partial differential equation which forms the basis of quantum mechanics [65–68] in modern physics. The general form of the time-dependent Schrödinger equation which gives a description of a system evolving with time is given as

$$\mathbf{j}\hbar\frac{\partial\Psi}{\partial t} = \hat{H}\Psi \quad (2.3.1)$$

where $\mathbf{j} = \sqrt{-1}$, $\hbar$ is the reduced Planck's constant (or Dirac constant), Ψ is the wave function of the quantum system and $\hat{H}$ is the Hamiltonian operator. The Hamiltonian operator, $\hat{H}$ involves the sum of the kinetic and potential energies which acts upon the wave function to generate the evolution of the wave function in time and space as shown:

$$\hat{H} = \hat{K} + \hat{\omega}(\vec{r}) \quad (2.3.2)$$

where $\hat{\omega}(\vec{r})$ is the potential energy function and

$$\hat{K} = \frac{p^2}{2m_e} \quad (2.3.3)$$

is the kinetic energy operator wherein m_e is the particle mass and p is the momentum operator. Since the momentum operator, p can be expressed as

$$p = -\mathbf{j}\hbar\nabla, \quad (2.3.4)$$

The Hamiltonian operator, $\hat{H}$ can now be written as

$$\hat{H} = -\frac{\hbar^2}{2m_e}\nabla^2 + \hat{\omega}(\vec{r}) \quad (2.3.5)$$

By substituting (2.3.5) into (2.3.1), the 2-D time-dependent Schrödinger equation

can now be expanded into

$$\begin{aligned} \mathbf{j}\hbar \frac{\partial \Psi(x, y, t)}{\partial t} &= \left[-\frac{\hbar^2}{2m_e} \nabla^2 + \hat{\omega}(x, y) \right] \Psi(x, y, t) \\ &= -\frac{\hbar^2}{2m_e} \frac{\partial^2 \Psi(x, y, t)}{\partial x^2} - \frac{\hbar^2}{2m_e} \frac{\partial^2 \Psi(x, y, t)}{\partial y^2} + \hat{\omega}(x, y) \Psi(x, y, t), \end{aligned} \quad (2.3.6)$$

with initial condition

$$\Psi(x, y, 0) = \phi(x, y), \quad (2.3.7)$$

and subject to the following boundary condition

$$\Lambda \frac{\partial \Psi(x, y, t)}{\partial n_\xi} + \Psi(x, y, t) = f(x, y, t), \quad t \geq 0 \quad (2.3.8)$$

where $\phi(x, y)$ and $f(x, y, t)$ are given functions, $\partial/\partial n_\xi$ is the differentiation along the outward direction normal to the boundary and Λ is a constant to determine the different types of boundary conditions. To obtain Dirichlet boundary condition [69], we can set $\Lambda = 0$, whereas Neumann boundary condition [70–72] can be obtained by setting $\Lambda = \infty$. For other finite values of $\Lambda \neq 0$, we will arrive at Robin boundary condition [73, 74].

2.3.1 Complex Explicit Method

In order to apply quantum FDTD method to the Schrödinger equation, a computation domain is established with grid spacing of Δx and Δy and time step Δt . Therefore, $\Psi(x, y, t)$ can be redefined as $\Psi(i\Delta x, j\Delta y, n\Delta t)$ for a particular time step n with grid point (i, j) , where i, j and n are integers. For simplicity, the notation $\Psi|_{i,j}^n$ is used to denote $\Psi(i\Delta x, j\Delta y, n\Delta t)$ for the quantum FDTD method.

Using the second order accurate central difference approximation for space and time derivatives, the explicit finite-difference approximation of the Schrödinger equation

can be obtained [75, 76], which forms the update equation given as

$$\begin{aligned} \Psi|_{i,j}^{n+1} = & \left(1 - \mathbf{j} \frac{2\widehat{\omega}_{i,j}\Delta t}{\hbar}\right) \Psi|_{i,j}^{n-1} + \mathbf{j} \frac{\hbar\Delta t}{m_e\Delta x^2} (\Psi|_{i+1,j,k}^n - 2\Psi|_{i,j,k}^n + \Psi|_{i-1,j,k}^n) \\ & + \mathbf{j} \frac{\hbar\Delta t}{m_e\Delta y^2} (\Psi|_{i,j+1,k}^n - 2\Psi|_{i,j,k}^n + \Psi|_{i,j-1,k}^n) \end{aligned} \quad (2.3.9)$$

This method has a second order accuracy in both time and space (error terms $O(\Delta\xi^3)$ and $O(\Delta t^3)$). Note that due to the complex wave function Ψ , the computation is rather taxing and time consuming. This is because the complex arithmetic operations in CPU are much slower than real arithmetic operations.

Due to the explicit nature of the quantum FDTD, it is subjected to a stability constraint [77, 78] given by

$$\Delta t \leq \frac{\hbar}{\frac{2\hbar^2}{m_e\Delta x^2} + \frac{2\hbar^2}{m_e\Delta y^2} + \widehat{\omega}_{i,j}} \quad (2.3.10)$$

which imposes a maximum constraint on its time step size. It can be seen that the maximum allowed time step is restricted not only by the smallest spatial step, but also the maximum value of the potential function. This limits the efficiency of the explicit quantum FDTD implementation. Furthermore, if the potential function is singular, it will render the computation of the Schrödinger equation to become unstable. Therefore, precautions have to be taken while discretizing the domain into grids using the quantum FDTD method to avoid the singular potential function.

2.3.2 Staggered Explicit Method

In 1991, Visscher [79] proposed a staggered explicit method of the time-dependent Schrödinger equation. It involves splitting of the complex wave function Ψ into two real functions that corresponds to the real and imaginary components [80–83] such

that

$$\Psi|_{i,j} = \Psi_R|_{i,j} + \mathbf{j}\Psi_I|_{i,j} \quad (2.3.11)$$

The time-dependent Schrödinger equation can then be updated using the two equations involving real functions corresponding to Ψ_R and Ψ_I

$$\begin{aligned} \Psi_R|_{i,j}^{n+1} = & \Psi_R|_{i,j}^n - \frac{\hbar\Delta t}{2m_e\Delta x^2} \left(\Psi_I|_{i+1,j}^{n+\frac{1}{2}} - 2\Psi_I|_{i,j}^{n+\frac{1}{2}} + \Psi_I|_{i-1,j}^{n+\frac{1}{2}} \right) \\ & - \frac{\hbar\Delta t}{2m_e\Delta y^2} \left(\Psi_I|_{i,j+1}^{n+\frac{1}{2}} - 2\Psi_I|_{i,j}^{n+\frac{1}{2}} + \Psi_I|_{i,j-1}^{n+\frac{1}{2}} \right) + \frac{\hat{\omega}_{i,j}\Delta t}{\hbar} \Psi_I|_{i,j}^{n+\frac{1}{2}} \end{aligned} \quad (2.3.12a)$$

$$\begin{aligned} \Psi_I|_{i,j}^{n+\frac{3}{2}} = & \Psi_I|_{i,j}^{n+\frac{1}{2}} + \frac{\hbar\Delta t}{2m_e\Delta x^2} \left(\Psi_R|_{i+1,j}^{n+1} - 2\Psi_R|_{i,j}^{n+1} + \Psi_R|_{i-1,j}^{n+1} \right) \\ & + \frac{\hbar\Delta t}{2m_e\Delta y^2} \left(\Psi_R|_{i,j+1}^{n+1} - 2\Psi_R|_{i,j}^{n+1} + \Psi_R|_{i,j-1}^{n+1} \right) - \frac{\hat{\omega}_{i,j}\Delta t}{\hbar} \Psi_R|_{i,j}^{n+1} \end{aligned} \quad (2.3.12b)$$

This method has third order error terms $O(\Delta\xi^3)$ and $O(\Delta t^3)$, having the same accuracy as the complex explicit method in (2.3.9). In the complex explicit method in (2.3.9), the real and imaginary components of the wave function are obtained in the same time step; whereas the staggered explicit method in (2.3.12) has the real and imaginary components of the wave function evaluated at half time step away.

The staggered explicit method is also subjected to a stability constraint [84,85] given by

$$\Delta t \leq \frac{\hbar}{\frac{\hbar^2}{m_e\Delta x^2} + \frac{\hbar^2}{m_e\Delta y^2} + \frac{\hat{\omega}_{i,j}}{2}} \quad (2.3.13)$$

However, by comparing (2.3.13) with (2.3.10), it is evident that with the same grid spacing and when the potential function $\hat{\omega}$ is zero, the maximum constraint on its time step size for the staggered explicit method is twice that of the complex explicit method. This in turn improves the efficiency of the staggered explicit method by two times.

2.3.3 Tridiagonal ADI (Tri-ADI) Method

To overcome stability constraints in (2.3.10) and (2.3.13) imposed by the complex and staggered explicit methods, unconditionally stable implicit algorithms have been introduced. The unconditionally stable CN based implicit quantum FDTD method [86–90] has been introduced. However, due to its implementation complexity, solving the updating equation is rather tedious and computationally intensive.

Another unconditionally stable ADI method has been introduced by Peaceman-Rachford [61]. Based on this ADI method, the updating equations are split into two procedures in the following [91, 92]:

$$\left(1 - \frac{\Delta t}{2} A_Q\right) \Psi^{n+\frac{1}{2}} = \left(1 + \frac{\Delta t}{2} B_Q\right) \Psi^n \quad (2.3.14a)$$

$$\left(1 - \frac{\Delta t}{2} B_Q\right) \Psi^{n+1} = \left(1 + \frac{\Delta t}{2} A_Q\right) \Psi^{n+\frac{1}{2}} \quad (2.3.14b)$$

where

$$A_Q = \mathbf{j} \frac{\hbar}{2m_e} \frac{\partial^2}{\partial x^2} - \mathbf{j} \frac{\hat{\omega}_{i,j}}{2\hbar} \quad \text{and} \quad B_Q = \mathbf{j} \frac{\hbar}{2m_e} \frac{\partial^2}{\partial y^2} - \mathbf{j} \frac{\hat{\omega}_{i,j}}{2\hbar}.$$

This method has an accuracy of second-order in both time and space (error terms $O(\Delta\xi^3)$ and $O(\Delta t^3)$). It has a tridiagonal system of equations which can be solved efficiently through Thomas algorithm.

By discretizing using central approximation for (2.3.14), we can obtain

$$\begin{aligned} -\mathbf{j}\hat{a}_{2x} \left(\Psi|_{i-1,j}^{n+\frac{1}{2}} + \Psi|_{i+1,j}^{n+\frac{1}{2}} \right) + \left(1 + \mathbf{j}\hat{a}_{1x} + \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^{n+\frac{1}{2}} \\ = \mathbf{j}\hat{a}_{2y} \left(\Psi|_{i,j-1}^n + \Psi|_{i,j+1}^n \right) + \left(1 - \mathbf{j}\hat{a}_{1y} - \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^n \end{aligned} \quad (2.3.15a)$$

$$\begin{aligned} -\mathbf{j}\hat{a}_{2y} \left(\Psi|_{i,j-1}^{n+1} + \Psi|_{i,j+1}^{n+1} \right) + \left(1 + \mathbf{j}\hat{a}_{1y} + \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^{n+1} \\ = \mathbf{j}\hat{a}_{2x} \left(\Psi|_{i-1,j}^{n+\frac{1}{2}} + \Psi|_{i+1,j}^{n+\frac{1}{2}} \right) + \left(1 - \mathbf{j}\hat{a}_{1x} - \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^{n+\frac{1}{2}} \end{aligned} \quad (2.3.15b)$$

where $\hat{a}_{\zeta\zeta} = \frac{\hbar\Delta t}{2\zeta m_e \Delta\xi^2}$, $\hat{b}_{i,j} = \frac{\Delta t \hat{\omega}_{i,j}}{4\hbar}$ and ζ is an integer.

It can be seen that the LHS of (2.3.15) constitutes the tridiagonal system of equations which shall be known as the Tri-ADI method. However, due to the nature of the complex wave function Ψ which involves real and imaginary parts, this algorithm is computationally expensive. In addition, the operators on the RHS of the equations further diminish the efficiency of this algorithm. Apart from CN and ADI methods, there are other implicit methods which also have unconditionally stable feature such as the SS method [93–95], but they shall not be discussed further in the thesis.

From Sections 2.1 to 2.3, the partial differential equations involved, namely Maxwell's equations, heat transfer equation and Schrödinger equation; required different types of stability constraint for their respective explicit methods. Since all these explicit methods can be converted into the implicit schemes of ADI and/or LOD methods, this will ensure the unconditionally stability of the partial differential equations.

Chapter 3

Development of Stable and Efficient ADI/LOD-FDTD Methods for Electromagnetics

3.1 Introduction

Recently, there has been increasing interest in the development of alternative unconditionally stable FDTD methods that are not constrained by the CFL condition, particularly on the ADI- [8–11], SS- [12–14] and LOD-FDTD methods [15,16]. However, this comes at an expense of increasing complexity in its implementation. Besides having to solve the tridiagonal systems, there are substantial amount of arithmetic operations and field variables involved on the RHS of the update equations, not to mention the huge amount of memory indexing operations incurred.

Based on the principle of fundamental implicit schemes [96,97], an efficient algorithm has been developed for both ADI- and LOD-FDTD methods. This features similar fundamental updating structures with matrix-operator-free RHS. The efficient fundamental ADI-FDTD and efficient fundamental LOD-FDTD, or FADI-FDTD and FLOD-FDTD in short, result in much simpler and more concise update equations than the conventional ADI- and LOD-FDTD implementations.

Nevertheless, despite having a more efficient and simpler implementation using the FADI- and FLOD-FDTD methods, continuing efforts are still being made to further increase the overall efficiency. Of late, programmable GPUs with highly parallel processors have led to the interest in using GPUs for general purpose programming [17–20, 98–100]. Such highly parallel processing feature of the GPU further motivates us into exploring the implementation of the fundamental implicit FDTD methods.

There is an increasing interest in artificial PMC surfaces [101] for antenna applications. In addition, a combination of both PMC and PEC boundary conditions called the perfect electromagnetic conductor (PEMC) boundary condition [102, 103] has also been introduced. For closed region simulation using the ADI- or LOD-FDTD methods, the implicit updating of electric fields facilitates the implementation of the PEC boundary condition. Likewise, the implicit updating of magnetic fields enables the convenient implementation of the PMC boundary condition. However, it is not too convenient when one would want to implement the PMC boundary condition for implicit electric fields or conversely, the PEC boundary condition for implicit magnetic fields.

To simulate the unbounded region problem, it requires the deployment of ABCs, in order to truncate the computation domain. Generally, there are two types of ABCs. One is the usage of outgoing wave equation such as the Mur's ABC [33, 34, 38]. This approach is simpler for the implementation with low memory consumption. However, the absorption is low (typically around -20dB), therefore computation results are not very accurate. For better absorption (typically around -60dB), nonphysical absorbing media such as the PML [39–42, 45–48] can be introduced. Although this approach is more memory-intensive and requires huge amount of memory indexing operations, it can absorb wide-angle scattered wave perfectly, resulting in highly

accurate computation results.

In this chapter, we will first formulate the conventional ADI- and LOD-FDTD methods into their respective efficient fundamental forms. Subsequently, we present the implementations of PMC and PEC boundary conditions for FADI- and FLOD-FDTD methods for closed region simulation. The PMC (or PEC) boundary equations for the implicit updating of electric (or magnetic) fields in FADI- and FLOD-FDTD methods are derived using image theory. Image theory facilitates the implementation of PMC and/or PEC boundary conditions regardless of whether the update equations for electric or magnetic fields are implicit. Comparisons between the PMC and PEC boundary equations in the conventional and efficient fundamental ADI- and LOD-FDTD methods based on flops count will be discussed in the following sections.

For open structure simulation, we present the implementation of the Mur first order ABC [33] in 3-D FADI- and FLOD-FDTD methods. The Mur ABC is incorporated into FADI- and FLOD-FDTD methods using consistent implementation and a novel implementation with lower reflection coefficient. The reflection coefficients for both implementations in FADI- and FLOD-FDTD methods are compared and validated with the conventional ADI- and LOD-FDTD methods for various Courant-Friedrich-Lewy number (CFLN). Efficiency gain in terms of the CPU computation time for both conventional and efficient fundamental ADI-FDTD methods will be analyzed.

For even better absorption, we present the implementation of the split-field PML [39] for the FADI-FDTD method. The split-field PML is formulated into the ADI generalized splitting formulae cast into matrix-operator-free RHS. The FADI-FDTD method with PML incorporated is compared to the conventional ADI-FDTD implementation through numerical simulations. It can be shown that their reflection errors are identical and confirmed that both implementations are equivalently ef-

fective. On the other hand, the efficiency gain of the FADI-FDTD method over conventional ADI-FDTD method with PML will be illustrated by comparing their CPU computation time.

Lastly, we present the GPU-accelerated FADI-FDTD method with CFS-CPML. The conventional ADI-FDTD method with CFS-CPML [47] is first cast into the compact matrix form and formulated into the FADI-FDTD method leading to a reduction of the update coefficients and field variables. By using CUDA, we further incorporate the FADI-FDTD method with CFS-CPML into the GPU to exploit data parallelism. Numerical results of reflection error and efficiency gain will be discussed. To demonstrate the usefulness of the FADI-FDTD method with CFS-CPML, a practical microstrip low-pass filter will be presented in the following sections. A high computational power can be attained while preserving a good agreement with the Yee-FDTD method.

3.2 Efficient Fundamental ADI- and LOD-FDTD Methods

In this section, the derivation of both FADI- and FLOD-FDTD methods will be described. Techniques to further simplify the updating equations are illustrated.

3.2.1 FADI-FDTD Method

We first derive the efficient fundamental form of the ADI-FDTD method. The conventional ADI-FDTD method can be implemented more conveniently with the introduction of an auxiliary variable to denote the RHS of the implicit updating equations. Now, (2.1.6) can be written as

$$\mathbf{v}_E^n = \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{B}_E \right) \mathbf{u}^n \quad (3.2.1a)$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^{n+\frac{1}{2}} = \mathbf{v}_E^n \quad (3.2.1b)$$

$$\mathbf{v}_E^{n+\frac{1}{2}} = \left(\mathbf{I}_6 + \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^{n+\frac{1}{2}} \quad (3.2.1c)$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{B}_E\right)\mathbf{u}^{n+1} = \mathbf{v}_E^{n+\frac{1}{2}} \quad (3.2.1d)$$

where

$$\mathbf{v}_E = [e_x, e_y, e_z, h_x, h_y, h_z]^T. \quad (3.2.2)$$

Note that $\mathbf{u}$ consists of electric (E) and magnetic (H) field components [c.f (2.1.7)] that refers to the actual field values. On the other hand, $\mathbf{v}_E$ consists of temporary auxiliary electric (e) and magnetic (h) field variables which do not require additional memory. The time-indexing for $\mathbf{v}_E$ has no implication with the update equations or accuracy of the overall equation.

By exploiting these auxiliary variables, the conventional ADI algorithm can be modified into the efficient scheme. First, we alter (3.2.1d) by one time step backward to get

$$\mathbf{v}_E^{n-\frac{1}{2}} = \left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{B}_E\right)\mathbf{u}^n. \quad (3.2.3)$$

With further manipulation of (3.2.1a) into

$$\begin{aligned} \mathbf{v}_E^n &= \left(\mathbf{I}_6 + \frac{\Delta t}{2}\mathbf{B}_E\right)\mathbf{u}^n \\ &= \mathbf{u}^n + \frac{\Delta t}{2}\mathbf{B}_E\mathbf{u}^n \\ &= 2\mathbf{u}^n - \mathbf{u}^n + \frac{\Delta t}{2}\mathbf{B}_E\mathbf{u}^n \\ &= 2\mathbf{u}^n - \left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{B}_E\right)\mathbf{u}^n. \end{aligned} \quad (3.2.4)$$

By substituting (3.2.3) into (3.2.4), we can obtain

$$\mathbf{v}_E^n = 2\mathbf{u}^n - \mathbf{v}_E^{n-\frac{1}{2}}. \quad (3.2.5)$$

Similarly, through (3.2.1b), we are able to reduce (3.2.1c) into

$$\begin{aligned}
\mathbf{v}_E^{n+\frac{1}{2}} &= \left(\mathbf{I}_6 + \frac{\Delta t}{2} \mathbf{A}_E \right) \mathbf{u}^{n+\frac{1}{2}} \\
&= 2\mathbf{u}^{n+\frac{1}{2}} - \left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{A}_E \right) \mathbf{u}^{n+\frac{1}{2}} \\
&= 2\mathbf{u}^{n+\frac{1}{2}} - \mathbf{v}_E^{n+\frac{1}{2}}.
\end{aligned} \tag{3.2.6}$$

By combining (3.2.5) and (3.2.6), we have simplified the algorithm in (3.2.1) to

$$\mathbf{v}_E^n = 2\mathbf{u}^n - \mathbf{v}_E^{n-\frac{1}{2}} \tag{3.2.7a}$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{A}_E \right) \mathbf{u}^{n+\frac{1}{2}} = \mathbf{v}_E^n \tag{3.2.7b}$$

$$\mathbf{v}_E^{n+\frac{1}{2}} = 2\mathbf{u}^{n+\frac{1}{2}} - \mathbf{v}_E^n \tag{3.2.7c}$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2} \mathbf{B}_E \right) \mathbf{u}^{n+1} = \mathbf{v}_E^{n+\frac{1}{2}}. \tag{3.2.7d}$$

With re-definition of the field variables

$$\tilde{\mathbf{u}}^n = 2\mathbf{u}^n, \quad \tilde{\mathbf{u}}^{n+\frac{1}{2}} = 2\mathbf{u}^{n+\frac{1}{2}}, \quad \tilde{\mathbf{u}}^{n+1} = 2\mathbf{u}^{n+1} \tag{3.2.8}$$

where

$$\tilde{\mathbf{u}} = [\tilde{E}_x, \tilde{E}_y, \tilde{E}_z, \tilde{H}_x, \tilde{H}_y, \tilde{H}_z]^T \tag{3.2.9}$$

Lastly, we obtain the final update procedures as

$$\mathbf{v}_E^n = \tilde{\mathbf{u}}^n - \mathbf{v}_E^{n-\frac{1}{2}} \tag{3.2.10a}$$

$$\left(\frac{1}{2} \mathbf{I}_6 - \frac{\Delta t}{4} \mathbf{A}_E \right) \tilde{\mathbf{u}}^{n+\frac{1}{2}} = \mathbf{v}_E^n \tag{3.2.10b}$$

$$\mathbf{v}_E^{n+\frac{1}{2}} = \tilde{\mathbf{u}}^{n+\frac{1}{2}} - \mathbf{v}_E^n \tag{3.2.10c}$$

$$\left(\frac{1}{2} \mathbf{I}_6 - \frac{\Delta t}{4} \mathbf{B}_E \right) \tilde{\mathbf{u}}^{n+1} = \mathbf{v}_E^{n+\frac{1}{2}} \tag{3.2.10d}$$

with initialization

$$\mathbf{v}_E^{-\frac{1}{2}} = \left(\frac{1}{2} \mathbf{I}_6 - \frac{\Delta t}{4} \mathbf{B}_E \right) \tilde{\mathbf{u}}^0. \tag{3.2.11}$$

The FADI algorithm in (3.2.10) has now been reduced into a simpler form without matrix operators $\mathbf{A}_E$ or $\mathbf{B}_E$ on the RHS. This results in a more concise update equation, leading to programming ease and better computational efficiency. The word fundamental is used to aptly describe the ADI-FDTD method in the fundamental (basic) form that cannot be reduced further on the RHS.

Expanding (3.2.10), upon some manipulations and arrangements, the full update equations of the FADI-FDTD method are given as follows:

For first procedure from n to $n + \frac{1}{2}$:

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j,k}^n = \tilde{E}_x|_{i+\frac{1}{2},j,k}^n - e_x|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} \quad (3.2.12a)$$

$$e_y|_{i,j+\frac{1}{2},k}^n = \tilde{E}_y|_{i,j+\frac{1}{2},k}^n - e_y|_{i,j+\frac{1}{2},k}^{n-\frac{1}{2}} \quad (3.2.12b)$$

$$e_z|_{i,j,k+\frac{1}{2}}^n = \tilde{E}_z|_{i,j,k+\frac{1}{2}}^n - e_z|_{i,j,k+\frac{1}{2}}^{n-\frac{1}{2}} \quad (3.2.12c)$$

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n = \tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n-\frac{1}{2}} \quad (3.2.12d)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n = \tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n-\frac{1}{2}} \quad (3.2.12e)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n = \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n-\frac{1}{2}} \quad (3.2.12f)$$

b) Implicit update for $\tilde{E}$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} \right) \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2} \left(\tilde{E}_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} + \tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} \right) \\ & = e_x|_{i+\frac{1}{2},j,k}^n + \frac{\Delta t}{2\epsilon\Delta y} \left(h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n \right) \end{aligned} \quad (3.2.13a)$$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} \right) \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta z^2} \left(\tilde{E}_y|_{i,j+\frac{1}{2},k-1}^{n+\frac{1}{2}} + \tilde{E}_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} \right) \\ & = e_y|_{i,j+\frac{1}{2},k}^n + \frac{\Delta t}{2\epsilon\Delta z} \left(h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - h_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n \right) \end{aligned} \quad (3.2.13b)$$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta x^2} \right) \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta x^2} \left(\tilde{E}_z|_{i-1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \tilde{E}_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \\ & = e_z|_{i,j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\epsilon\Delta x} \left(h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - h_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n \right). \end{aligned} \quad (3.2.13c)$$

c) Explicit update for $\tilde{H}$

$$\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = 2h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta z} \left(\tilde{E}_y|_{i,j+\frac{1}{2},k+1}^n - \tilde{E}_y|_{i,j+\frac{1}{2},k}^n \right) \quad (3.2.14a)$$

$$\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = 2h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta x} \left(\tilde{E}_z|_{i+1,j,k+\frac{1}{2}}^n - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^n \right) \quad (3.2.14b)$$

$$\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = 2h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\Delta t}{2\mu\Delta y} \left(\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^n - \tilde{E}_x|_{i+\frac{1}{2},j,k}^n \right). \quad (3.2.14c)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - e_x|_{i+\frac{1}{2},j,k}^n \quad (3.2.15a)$$

$$e_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - e_y|_{i,j+\frac{1}{2},k}^n \quad (3.2.15b)$$

$$e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - e_z|_{i,j,k+\frac{1}{2}}^n \quad (3.2.15c)$$

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n \quad (3.2.15d)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n \quad (3.2.15e)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n \quad (3.2.15f)$$

Note that if the explicit update for $\tilde{H}^{n+\frac{1}{2}}$ is not required, we can substitute (3.2.14)

into the auxiliary (explicit) update of $h^{n+\frac{1}{2}}$ in (3.2.15d)-(3.2.15f) as

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta z} \left(\tilde{E}_y|_{i,j+\frac{1}{2},k+1}^n - \tilde{E}_y|_{i,j+\frac{1}{2},k}^n \right) \quad (3.2.16a)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta x} \left(\tilde{E}_z|_{i+1,j,k+\frac{1}{2}}^n - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^n \right) \quad (3.2.16b)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\Delta t}{2\mu\Delta y} \left(\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^n - \tilde{E}_x|_{i+\frac{1}{2},j,k}^n \right) \quad (3.2.16c)$$

b) Implicit update for $\tilde{E}$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} \right) \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} - \frac{\Delta t^2}{8\epsilon\mu\Delta z^2} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k-1}^{n+1} + \tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} \right) \\ & = e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\epsilon\Delta z} \left(h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - h_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (3.2.17a)$$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta x^2} \right) \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} - \frac{\Delta t^2}{8\epsilon\mu\Delta x^2} \left(\tilde{E}_y|_{i-1,j+\frac{1}{2},k}^{n+1} + \tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} \right) \\ &= e_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\epsilon\Delta x} \left(h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - h_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \end{aligned} \quad (3.2.17b)$$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} \right) \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2} \left(\tilde{E}_z|_{i,j-1,k+\frac{1}{2}}^{n+1} + \tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^{n+1} \right) \\ &= e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\epsilon\Delta y} \left(h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - h_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right). \end{aligned} \quad (3.2.17c)$$

c) Explicit update for $\tilde{H}$

$$\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = 2h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta y} \left(\tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (3.2.18a)$$

$$\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = 2h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta z} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (3.2.18b)$$

$$\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = 2h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right). \quad (3.2.18c)$$

For non-zero initial $\tilde{E}^0$ and $\tilde{H}^0$, we apply the initialization as follows:

$$e_x|_{i+\frac{1}{2},j,k}^{-\frac{1}{2}} = \frac{1}{2} \tilde{E}_x|_{i+\frac{1}{2},j,k}^0 + \frac{\Delta t}{4\epsilon\Delta z} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^0 - \tilde{H}_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^0 \right) \quad (3.2.19a)$$

$$e_y|_{i,j+\frac{1}{2},k}^{-\frac{1}{2}} = \frac{1}{2} \tilde{E}_y|_{i,j+\frac{1}{2},k}^0 + \frac{\Delta t}{4\epsilon\Delta x} \left(\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^0 - \tilde{H}_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^0 \right) \quad (3.2.19b)$$

$$e_z|_{i,j,k+\frac{1}{2}}^{-\frac{1}{2}} = \frac{1}{2} \tilde{E}_z|_{i,j,k+\frac{1}{2}}^0 + \frac{\Delta t}{4\epsilon\Delta y} \left(\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^0 - \tilde{H}_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^0 \right) \quad (3.2.19c)$$

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{-\frac{1}{2}} = \frac{1}{2} \tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^0 + \frac{\Delta t}{4\mu\Delta y} \left(\tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^0 - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^0 \right) \quad (3.2.19d)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{-\frac{1}{2}} = \frac{1}{2} \tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^0 + \frac{\Delta t}{4\mu\Delta z} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k+1}^0 - \tilde{E}_x|_{i+\frac{1}{2},j,k}^0 \right) \quad (3.2.19e)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{-\frac{1}{2}} = \frac{1}{2} \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^0 + \frac{\Delta t}{4\mu\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^0 - \tilde{E}_y|_{i,j+\frac{1}{2},k}^0 \right) \quad (3.2.19f)$$

Similarly, if the explicit update for $\tilde{H}^{n+1}$ is not required, we can substitute (3.2.18) into the auxiliary (explicit) update of h^{n+1} in (3.2.12d)-(3.2.12f) as

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta y} \left(\tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (3.2.20a)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta z} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (3.2.20b)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (3.2.20c)$$

Table 3.1: Flops count for the conventional and efficient fundamental ADI-FDTD methods

Scheme		ADI (2.1.6)	FADI (3.2.10)
Implicit	A/S	48	12
	M/D	18	6
Explicit	A/S	24	18
	M/D	12	6
Total	A/S	72	30
	M/D	30	12
	A/S + M/D	102	42

By comparing (3.2.12)-(3.2.18) with (2.1.18)-(2.1.19), it can be seen that there is a decrease in the number of arithmetic operations and memory indexing. To quantify the reduction in the number of arithmetic operation (in terms of flops count), the number of additions/subtractions (A/S) and multiplications/divisions (M/D) required is determined. Table 3.1 shows the flops count for the conventional and efficient fundamental ADI-FDTD methods. The comparison of these update equations is based on their respective RHS. The conventional ADI-FDTD method requires a total flops count (A/S+M/D) of 72 while the FADI-FDTD method only requires a total flop count of 42. This results in an improvement of the overall efficiency. It is worth noting that the reduction of computation time is more significant for M/D as compared to A/S. The total flops count (A/S+M/D) is to provide the readers a better insight and for comparison purposes. Although the actual flops count has been taken into account, some other factors still exist that may affect the actual computation efficiency. These factors are often dependent on the particular computer platform, hardware configuration, operating system, compiler software, program code arrangement, etc.

3.2.2 FLOD-FDTD Method

Next, we derive the efficient fundamental form of the LOD-FDTD method. The conventional LOD-FDTD method can be modified with its RHS free of matrix operators. From (2.1.20a), we have

$$\begin{aligned} \left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^{n+\frac{1}{2}} &= \left(\mathbf{I}_6 + \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^n \\ &= 2\mathbf{u}^n - \left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{A}_E\right)\mathbf{u}^n \end{aligned} \quad (3.2.21)$$

Through arithmetic manipulations, we obtain

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}\mathbf{A}_E\right)\left(\mathbf{u}^{n+\frac{1}{2}} + \mathbf{u}^n\right) = 2\mathbf{u}^n \quad (3.2.22)$$

By introducing an auxiliary variable $\mathbf{v}_E$, the vector terms $\left(\mathbf{u}^{n+\frac{1}{2}} + \mathbf{u}^n\right)$ are denoted as

$$\mathbf{v}_E^{n+\frac{1}{2}} = \mathbf{u}^{n+\frac{1}{2}} + \mathbf{u}^n \quad (3.2.23)$$

By applying similar manipulations for (2.1.20b) and combining all field and auxiliary variables, we obtain the FLOD-FDTD method written as

$$\left(\frac{1}{2}\mathbf{I}_6 - \frac{\Delta t}{4}\mathbf{A}_E\right)\mathbf{v}_E^{n+\frac{1}{2}} = \mathbf{u}^n \quad (3.2.24a)$$

$$\mathbf{u}^{n+\frac{1}{2}} = \mathbf{v}_E^{n+\frac{1}{2}} - \mathbf{u}^n \quad (3.2.24b)$$

$$\left(\frac{1}{2}\mathbf{I}_6 - \frac{\Delta t}{4}\mathbf{B}_E\right)\mathbf{v}_E^{n+1} = \mathbf{u}^{n+\frac{1}{2}} \quad (3.2.24c)$$

$$\mathbf{u}^{n+1} = \mathbf{v}_E^{n+1} - \mathbf{u}^{n+\frac{1}{2}}. \quad (3.2.24d)$$

Similar to the FADI-FDTD, the FLOD algorithm has now been reduced into a simpler form with matrix-operator-free RHS. However, it can be seen that there is no initialization required for the FLOD-FDTD method and the resultant field solution can be obtained directly from (3.2.24d).

By expanding (3.2.24), with further modifications, the full update equations of the FLOD-FDTD method are shown below

For first procedure from n to $n + \frac{1}{2}$:

a) Auxiliary implicit update for e

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\right) e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2} \left(e_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}}\right) \\ = E_x|_{i+\frac{1}{2},j,k}^n + \frac{\Delta t}{2\epsilon\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) \end{aligned} \quad (3.2.25a)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta z^2}\right) e_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta z^2} \left(e_y|_{i,j+\frac{1}{2},k-1}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}}\right) \\ = E_y|_{i,j+\frac{1}{2},k}^n + \frac{\Delta t}{2\epsilon\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n\right) \end{aligned} \quad (3.2.25b)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta x^2}\right) e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta x^2} \left(e_z|_{i-1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}}\right) \\ = E_z|_{i,j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\epsilon\Delta x} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n\right). \end{aligned} \quad (3.2.25c)$$

b) Auxiliary explicit update for h

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = 2H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta z} \left(e_y|_{i,j+\frac{1}{2},k+1}^n - e_y|_{i,j+\frac{1}{2},k}^n\right) \quad (3.2.26a)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = 2H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta x} \left(e_z|_{i+1,j,k+\frac{1}{2}}^n - e_z|_{i,j,k+\frac{1}{2}}^n\right) \quad (3.2.26b)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = 2H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\Delta t}{2\mu\Delta y} \left(e_x|_{i+\frac{1}{2},j+1,k}^n - e_x|_{i+\frac{1}{2},j,k}^n\right). \quad (3.2.26c)$$

c) Explicit update for E and H

$$E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^n \quad (3.2.27a)$$

$$E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = e_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^n \quad (3.2.27b)$$

$$E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^n \quad (3.2.27c)$$

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n \quad (3.2.27d)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n \quad (3.2.27e)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n \quad (3.2.27f)$$

Note that the update equations in the first procedure can be further simplified by substituting the auxiliary explicit update for $h^{n+\frac{1}{2}}$ in (3.2.26) into the explicit update for $H^{n+\frac{1}{2}}$ in (3.2.27d)-(3.2.27f) as

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta z} \left(e_y|_{i,j+\frac{1}{2},k+1}^n - e_y|_{i,j+\frac{1}{2},k}^n \right) \quad (3.2.28a)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\Delta t}{2\mu\Delta x} \left(e_z|_{i+1,j,k+\frac{1}{2}}^n - e_z|_{i,j,k+\frac{1}{2}}^n \right) \quad (3.2.28b)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\Delta t}{2\mu\Delta y} \left(e_x|_{i+\frac{1}{2},j+1,k}^n - e_x|_{i+\frac{1}{2},j,k}^n \right). \quad (3.2.28c)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Auxiliary implicit update for e

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta z^2} \right) e_x|_{i+\frac{1}{2},j,k}^{n+1} - \frac{\Delta t^2}{8\epsilon\mu\Delta z^2} \left(e_x|_{i+\frac{1}{2},j,k-1}^{n+1} + e_x|_{i+\frac{1}{2},j,k+1}^{n+1} \right) \\ = E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\epsilon\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (3.2.29a)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta x^2} \right) e_y|_{i,j+\frac{1}{2},k}^{n+1} - \frac{\Delta t^2}{8\epsilon\mu\Delta x^2} \left(e_y|_{i-1,j+\frac{1}{2},k}^{n+1} + e_y|_{i+1,j+\frac{1}{2},k}^{n+1} \right) \\ = E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\epsilon\Delta x} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \end{aligned} \quad (3.2.29b)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} \right) e_z|_{i,j,k+\frac{1}{2}}^{n+1} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2} \left(e_z|_{i,j-1,k+\frac{1}{2}}^{n+1} + e_z|_{i,j+1,k+\frac{1}{2}}^{n+1} \right) \\ = E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\epsilon\Delta y} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right). \end{aligned} \quad (3.2.29c)$$

b) Auxiliary explicit update for h

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = 2H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta y} \left(e_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - e_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (3.2.30a)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = 2H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta z} \left(e_x|_{i+\frac{1}{2},j,k+1}^{n+1} - e_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (3.2.30b)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = 2H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta x} \left(e_y|_{i+1,j+\frac{1}{2},k}^{n+1} - e_y|_{i,j+\frac{1}{2},k}^{n+1} \right). \quad (3.2.30c)$$

c) Explicit update for E and H

$$E_x|_{i+\frac{1}{2},j,k}^{n+1} = e_x|_{i+\frac{1}{2},j,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \quad (3.2.31a)$$

$$E_y|_{i,j+\frac{1}{2},k}^{n+1} = e_y|_{i,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (3.2.31b)$$

Table 3.2: Flops count for the conventional and efficient fundamental LOD-FDTD methods

Scheme		LOD (2.1.20)	FLOD (3.2.24)
Implicit	A/S	24	12
	M/D	18	6
Explicit	A/S	24	18
	M/D	6	6
Total	A/S	48	42
	M/D	24	30
	A/S + M/D	72	42

$$E_z|_{i,j,k+\frac{1}{2}}^{n+1} = e_z|_{i,j,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (3.2.31c)$$

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} - H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (3.2.31d)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (3.2.31e)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (3.2.31f)$$

Similarly, the update equations in the second procedure can be further simplified by substituting the auxiliary explicit update for h^{n+1} in (3.2.30) into the explicit update for H^{n+1} in (3.2.31d)-(3.2.31f) as

$$H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta y} \left(e_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - e_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (3.2.32a)$$

$$H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta z} \left(e_x|_{i+\frac{1}{2},j,k+1}^{n+1} - e_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (3.2.32b)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t}{2\mu\Delta x} \left(e_y|_{i+1,j+\frac{1}{2},k}^{n+1} - e_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (3.2.32c)$$

Table 3.2 shows the flops count for the conventional and efficient fundamental LOD-FDTD methods. The comparison of these update equations are based on their respective RHS. By comparing the flops count of the conventional LOD-FDTD method

in (2.1.23)-(2.1.24) with the FLOD-FDTD method in (3.2.25)-(3.2.32), it can be seen that the number of arithmetic operations (in terms of flops count) is reduced from 72 (for the LOD-FDTD method) to 42 (for the FLOD-FDTD method). Therefore, a greater efficiency for the FLOD-FDTD method can be attained. Furthermore, through a substitution of variables h in (3.2.26) and (3.2.30) into variables H in (3.2.27d)-(3.2.27f) and (3.2.31d)-(3.2.31f) respectively, we can minimize the number of updating equations without compromising the output of H fields.

3.3 PMC and PEC Boundary Conditions for FDTD Methods

In this section, detailed implementations of PMC and PEC boundary conditions for both FADI- and FLOD-FDTD methods are presented. Numerical results of the conventional and efficient fundamental ADI- and LOD-FDTD methods with PMC and/or PEC will be demonstrated.

3.3.1 Boundary conditions for FADI-FDTD Method

PMC for implicit electric fields

For convenience, we rewrite the main grid update equations for E_x field in (3.2.12a) and (3.2.13a) for the first procedure from n to $n + \frac{1}{2}$:

$$e_x|_{i+\frac{1}{2},j,k}^n = \tilde{E}_x|_{i+\frac{1}{2},j,k}^n - e_x|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} \quad (3.3.1a)$$

$$\begin{aligned} & \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} \right) \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2} \left(\tilde{E}_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} + \tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} \right) \\ & = e_x|_{i+\frac{1}{2},j,k}^n + \frac{\Delta t}{2\epsilon\Delta y} \left(h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n \right). \end{aligned} \quad (3.3.1b)$$

When E fields are updated implicitly, the PEC boundary condition can be conveniently implemented. However, it is not too convenient to implement the PMC

boundary condition due to some additional (out-of-domain) terms in the tridiagonal matrix of E fields. Following the treatment in standard FDTD [104], these terms can be avoided through the application of image theory for the PMC implementation.

Let us consider the boundary at $y = -\frac{\Delta y}{2}$ being set to the PMC boundary condition with $H_z|_{i+\frac{1}{2}, -\frac{1}{2}, k} = 0$. Through the principle of image theory, we have

$$E_x|_{i+\frac{1}{2}, 0, k} = E_x|_{i+\frac{1}{2}, -1, k}. \quad (3.3.2)$$

By substituting (3.3.2) to (3.3.1b), we obtain the PMC boundary equations as

$$e_x|_{i+\frac{1}{2}, 0, k}^n = \tilde{E}_x|_{i+\frac{1}{2}, 0, k}^n - e_x|_{i+\frac{1}{2}, 0, k}^{n-\frac{1}{2}} \quad (3.3.3a)$$

$$\left(\frac{1}{2} + \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\right)\tilde{E}_x|_{i+\frac{1}{2}, 0, k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\tilde{E}_x|_{i+\frac{1}{2}, 1, k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2}, 0, k}^n + \frac{\Delta t}{2\epsilon\Delta y}h_z|_{i+\frac{1}{2}, \frac{1}{2}, k}^n. \quad (3.3.3b)$$

The FADI-FDTD method has the RHS of (3.3.3b) expressed in terms of auxiliary field variables with the number of arithmetic operations reduced to the minimal. Similar reductions can be achieved for the boundary equations in the second procedure from $n + \frac{1}{2}$ to $n + 1$, which are omitted here and henceforth since they can be derived following the same manner.

PEC for implicit magnetic fields

We consider another case when H fields are updated implicitly. Based on the FADI-FDTD method in (3.2.10), we can derive the main grid update equations for H_z field in the first procedure from n to $n + \frac{1}{2}$ as

$$h_z|_{i+\frac{1}{2}, j+\frac{1}{2}, k}^n = \tilde{H}_z|_{i+\frac{1}{2}, j+\frac{1}{2}, k}^n - h_z|_{i+\frac{1}{2}, j+\frac{1}{2}, k}^{n-\frac{1}{2}} \quad (3.3.4a)$$

$$\begin{aligned} &\left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\right)\tilde{H}_z|_{i+\frac{1}{2}, j+\frac{1}{2}, k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\left(\tilde{H}_z|_{i+\frac{1}{2}, j-\frac{1}{2}, k}^{n+\frac{1}{2}} + \tilde{H}_z|_{i+\frac{1}{2}, j+\frac{3}{2}, k}^{n+\frac{1}{2}}\right) \\ &= h_z|_{i+\frac{1}{2}, j+\frac{1}{2}, k}^n + \frac{\Delta t}{2\epsilon\Delta y}\left(e_x|_{i+\frac{1}{2}, j+1, k}^n - e_x|_{i+\frac{1}{2}, j, k}^n\right). \end{aligned} \quad (3.3.4b)$$

While the PMC boundary condition can be conveniently implemented in (3.3.4), some inconveniences may be encountered when one is to implement the PEC boundary condition. We put into consideration the boundary at $y = 0$ being set to the PEC boundary condition with $E_x|_{i+\frac{1}{2},0,k} = 0$. By applying the principle of image theory, we present

$$H_z|_{i+\frac{1}{2},\frac{1}{2},k} = H_z|_{i+\frac{1}{2},-\frac{1}{2},k}. \quad (3.3.5)$$

By substituting (3.3.5) to (3.3.4b), the update equations for the PEC boundary condition result in

$$h_z|_{i+\frac{1}{2},\frac{1}{2},k}^n = \tilde{H}_z|_{i+\frac{1}{2},\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},\frac{1}{2},k}^{n-\frac{1}{2}} \quad (3.3.6a)$$

$$\left(\frac{1}{2} + \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\right)\tilde{H}_z|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\tilde{H}_z|_{i+\frac{1}{2},\frac{3}{2},k}^{n+\frac{1}{2}} = h_z|_{i+\frac{1}{2},\frac{1}{2},k}^n + \frac{\Delta t}{2\epsilon\Delta y}e_x|_{i+\frac{1}{2},1,k}^n. \quad (3.3.6b)$$

It can be seen that for the FADI-FDTD method, image theory facilitates the implementation of PMC and/or PEC boundary conditions regardless of whether the update equations for E or H fields are implicit.

3.3.2 Boundary conditions for FLOD-FDTD Method

PMC for implicit electric fields

For the FLOD-FDTD method, we rewrite the main grid update equations for E_x field in (3.2.25a) and (3.2.27a) for the first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} &\left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\right)e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\left(e_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}}\right) \\ &= E_x|_{i+\frac{1}{2},j,k}^n + \frac{\Delta t}{2\epsilon\Delta y}\left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) \end{aligned} \quad (3.3.7a)$$

$$E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^n. \quad (3.3.7b)$$

The implicit updating of E fields allows the PEC boundary condition to be implemented easily. In order to enable the implicit updating of E fields while satisfying the PMC boundary condition, we apply image theory to (3.3.7a) and obtain

$$\left(\frac{1}{2} + \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\right)e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},0,k}^n + \frac{\Delta t}{2\epsilon\Delta y}H_z|_{i+\frac{1}{2},\frac{1}{2},k}^n \quad (3.3.8a)$$

$$E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},0,k}^n. \quad (3.3.8b)$$

The FLOD-FDTD method has the LHS of (3.3.8a) expressed in terms of auxiliary field variables. Furthermore, the number of arithmetic operations on the RHS of (3.3.8) has been reduced.

PEC for implicit magnetic fields

Based on the FLOD-FDTD method in (3.2.24), the main grid update equations for H_z field in the first procedure from n to $n + \frac{1}{2}$ can be derived as

$$\begin{aligned} \left(\frac{1}{2} + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\right)h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\left(h_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} + h_z|_{i+\frac{1}{2},j+\frac{3}{2},k}^{n+\frac{1}{2}}\right) \\ = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\Delta t}{2\epsilon\Delta y}\left(E_x|_{i+\frac{1}{2},j+1,k}^n - E_x|_{i+\frac{1}{2},j,k}^n\right) \end{aligned} \quad (3.3.9a)$$

$$H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n. \quad (3.3.9b)$$

By applying image theory to (3.3.9a), we obtain the update equations for the PEC boundary condition as

$$\left(\frac{1}{2} + \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}\right)h_z|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{8\epsilon\mu\Delta y^2}h_z|_{i+\frac{1}{2},\frac{3}{2},k}^{n+\frac{1}{2}} = H_z|_{i+\frac{1}{2},\frac{1}{2},k}^n + \frac{\Delta t}{2\epsilon\Delta y}E_x|_{i+\frac{1}{2},1,k}^n \quad (3.3.10a)$$

$$H_z|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} = h_z|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},\frac{1}{2},k}^n. \quad (3.3.10b)$$

Similar to the ADI-FDTD method, image theory facilitates the implementation of PMC and/or PEC boundary conditions for the FLOD-FDTD method, regardless of

Figure 3.1: Geometry with the shaded region representing either PMC or PEC boundary conditions.

whether the update equations for E or H fields are implicit.

3.3.3 Numerical Results

We perform numerical simulations for both conventional and efficient fundamental ADI- and LOD-FDTD methods with PMC and/or PEC boundary conditions. For reference, we provide the PMC boundary equations in the first procedure from n to $n + \frac{1}{2}$ for both conventional ADI- and LOD-FDTD methods as

a) Conventional ADI-FDTD:

$$\begin{aligned}
 & \left(1 + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\right) E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \\
 &= E_x|_{i+\frac{1}{2},0,k}^n + \frac{\Delta t}{2\epsilon\Delta y} H_z|_{i+\frac{1}{2},\frac{1}{2},k}^n - \frac{\Delta t^2}{4\epsilon\mu\Delta x\Delta y} \left(E_y|_{i+1,\frac{1}{2},k}^n - E_y|_{i,\frac{1}{2},k}^n\right) \\
 & \quad - \frac{\Delta t}{2\epsilon\Delta z} \left(H_y|_{i+\frac{1}{2},0,k+\frac{1}{2}}^n - H_y|_{i+\frac{1}{2},0,k-\frac{1}{2}}^n\right). \tag{3.3.11}
 \end{aligned}$$

Figure 3.2: Incident and reflected waveforms for (a) PMC boundary condition and (b) PEC boundary condition.

b) Conventional LOD-FDTD:

$$\begin{aligned}
 & \left(1 + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2}\right) E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} - \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \\
 &= E_x|_{i+\frac{1}{2},0,k}^n + \frac{\Delta t}{\epsilon\Delta y} H_z|_{i+\frac{1}{2},\frac{1}{2},k}^n + \frac{\Delta t^2}{4\epsilon\mu\Delta y^2} \left(E_x|_{i+\frac{1}{2},1,k}^n - E_x|_{i+\frac{1}{2},0,k}^n\right). \quad (3.3.12)
 \end{aligned}$$

Figure 3.1 shows the computation domain with dimension of $200 \times 120 \times 5$ grids and spatial step $\Delta x = \Delta y = \Delta z = 2.0$ mm. The boundary for the shaded region is either a PMC or PEC boundary condition. For the unshaded region, we apply the Mur first order ABC in Section 3.4 to absorb any reflection incurred. A line source excitation is located at 43 cells away from a PMC or PEC boundary. The source is

Table 3.3: Flops count for the PMC and PEC boundary equations

Scheme	ADI-FDTD			LOD-FDTD		
	A/S	M/D	Total	A/S	M/D	Total
Conventional (3.3.11), (3.3.12)	5	3	8	3	2	5
Efficient Fundamental (3.3.3), (3.3.8)	2	1	3	2	1	3

a differentiated Gaussian pulse applied to E_z component:

$$J_z = -2 \frac{(t - t_0)}{\tau} e^{-(\frac{t-t_0}{\tau})^2}, \tau = 40 \text{ ps}, t_0 = 3\tau. \quad (3.3.13)$$

An observation point is located 8 cells away from the source and 36 cells from the boundary.

Figure 3.2 plots the incident and reflected waveforms recorded for PMC and PEC boundary conditions using both conventional and efficient fundamental ADI- and LOD-FDTD methods. From Figure 3.2a, it can be seen that the polarity of the reflected E_z waveform from the PMC boundary is the same as the incident waveform. Meanwhile, from Figure 3.2b, the polarity of the reflected E_z waveform from the PEC boundary is opposite that of the incident waveform. This is consistent with the principle of image theory. In addition, we can observe that the numerical results of both conventional and efficient fundamental ADI- and LOD-FDTD methods are identical which confirm that both implementations are equivalently effective.

Table 3.3 shows the flops count for the PMC (and PEC) boundary equations in ADI- and LOD-FDTD methods. The comparison of these boundary equations in the conventional and efficient fundamental schemes is based on their respective RHS. From Table 3.3, it is justified that the total flops count (A/S+M/D) has been reduced considerably using the efficient fundamental schemes as compared to the

conventional counterparts. There is also a reduction in the total flops count for the explicit update equations in the efficient fundamental schemes (shown in Tables 3.1 and 3.2). However, we do not take them into consideration since they are not directly related to the boundary conditions. Overall they contribute together to the higher efficiency and simplicity for the efficient fundamental schemes.

3.4 Mur ABC for FDTD Methods

The previous section presented the implementation of PMC and PEC boundary conditions for FADI- and FLOD-FDTD methods in a closed region simulation. To simulate open structure, it requires the implementation of ABCs. For simplicity in implementation and low memory consumption, the Mur ABC is incorporated into FADI- and FLOD-FDTD methods. The two devised Mur ABC implementations for FADI- and FLOD-FDTD methods will be discussed in the following sections.

3.4.1 Mur ABC for ADI- and LOD-FDTD Methods

There are various measures to incorporate the Mur first order ABC into the ADI- and LOD-FDTD updating equations [36, 38]. Among them, we have devised two Mur ABC implementations for ADI- and LOD-FDTD methods as follows:

Consistent Implementation

The consistent implementation with the same time instant at both sides of the one-way wave equation (illustrated for E_x variable at the boundary $y = 0$) is as follows: For first procedure from n to $n + \frac{1}{2}$:

$$\frac{\partial}{\partial t} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{4}} = v \frac{\partial}{\partial y} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{4}} \quad (3.4.1a)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\frac{\partial}{\partial t} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{3}{4}} = v \frac{\partial}{\partial y} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{3}{4}} \quad (3.4.1b)$$

Novel Implementation

By varying the time instant of the one-way wave equation, we propose a novel implementation (illustrated for E_x variable at the boundary $y = 0$) as follows:

For first procedure from n to $n + 1/2$:

$$\frac{\partial}{\partial t} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{4}} = v \frac{\partial}{\partial y} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{2}} \quad (3.4.2a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\frac{\partial}{\partial t} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{3}{4}} = v \frac{\partial}{\partial y} E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+1}. \quad (3.4.2b)$$

It is worth noting that the novel implementation only adopts spatial averaging for its update equations, while the consistent implementation adopts both time and spatial averaging for its update equations. The novel implementation in (3.4.2) is a fully implicit, backward-time centered-space (BTCS) method [105]. As both ADI- and LOD-FDTD methods are also implicit schemes, the novel implementation is stable. Without using the same time step on both sides of the equations, the novel implementation in (3.4.2) has error terms $O(\Delta t^2)$ and $O(\Delta \xi^3)$. On the other hand, the consistent implementation (with same time step on both sides of the equations) in (3.4.1) has third order error terms $O(\Delta t^3)$ and $O(\Delta \xi^3)$. Since the novel or consistent implementation is only applied on the boundary of the computation domain, it does not have a significant effect on the accuracy of the whole domain based on the ADI-FDTD (with error terms $O(\Delta t^3)$ and $O(\Delta \xi^3)$) or LOD-FDTD (with error terms $O(\Delta t^2)$ and $O(\Delta \xi^3)$) method.

The equations in (3.4.1) and (3.4.2) are intermediate update equations for the Mur ABC. Time averaging (on the RHS of the update equations for consistent implementation) and finite-difference (on the LHS of the update equations for both implementations) are still required in order to achieve the final update equations. For the conventional 3-D ADI- and LOD-FDTD algorithms, the updating equations for E_x variable at boundaries $y = 0$ and $z = 0$ are discretized as follows:

i) Consistent Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \\ = \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^n + \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^n \end{aligned} \quad (3.4.3a)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},j,1}^n + \frac{2\Delta z - v\Delta t}{2\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,0}^n - E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \right) \quad (3.4.3b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+1} \\ = \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \end{aligned} \quad (3.4.4a)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},0,k}^{n+1} = E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} + \frac{2\Delta y - v\Delta t}{2\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},1,k}^{n+1} \right) \quad (3.4.4b)$$

ii) Novel Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},0,k}^n + E_x|_{i+\frac{1}{2},1,k}^n \quad (3.4.5a)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{\Delta z}{\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,0}^n + E_x|_{i+\frac{1}{2},j,1}^n \right) \quad (3.4.5b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+1} = E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (3.4.6a)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{\Delta y}{\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \right) \quad (3.4.6b)$$

The complete update equations of the Mur ABC for both consistent and novel implementations in the conventional 3-D ADI- and LOD-FDTD algorithms are in Appendix C. It is worth noticing that the Mur ABC cannot be directly incorporated into the efficient fundamental schemes, due to the formulations of FADI- and FLOD-FDTD methods which involve auxiliary variables. Apart from above, other Mur ABC implementation has been introduced into the conventional LOD-FDTD method, e.g. [106]. However, the conditions [106, (6), (9)] seem to be valid only for the 2-D transverse magnetic (TM) case and it is not clear how they are to extend for the 3-D LOD-FDTD method. Hence such implementation is omitted. Subsequently, detailed formulations of the Mur ABC into FADI- and FLOD-FDTD methods will be discussed in Sections 3.4.2 and 3.4.3 respectively.

3.4.2 Mur ABC for FADI-FDTD Method

To incorporate the Mur ABC into the FADI-FDTD method, we first consider the consistent implementation.

Consistent Implementation

By applying central difference approximation and time averaging in (3.4.1), we get

For first procedure from n to $n + 1/2$:

$$\left(1 - \frac{v\Delta t}{4} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{2}} = \left(1 + \frac{v\Delta t}{4} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^n \quad (3.4.7a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\left(1 - \frac{v\Delta t}{4} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+1} = \left(1 + \frac{v\Delta t}{4} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{2}}. \quad (3.4.7b)$$

We then formulate the Mur ABC for all E fields into the compact matrix form as

$$\left(\begin{bmatrix} \square & 0 \\ 0 & \square \end{bmatrix} - \frac{\Delta t}{2} \begin{bmatrix} \hat{\mathbf{C}} & 0 \\ 0 & -\hat{\mathbf{C}} \end{bmatrix}\right) \begin{bmatrix} \mathbf{q}_{min}^{n+\frac{1}{2}} \\ \mathbf{q}_{max}^{n+\frac{1}{2}} \end{bmatrix} = \left(\begin{bmatrix} \square & 0 \\ 0 & \square \end{bmatrix} + \frac{\Delta t}{2} \begin{bmatrix} \hat{\mathbf{C}} & 0 \\ 0 & -\hat{\mathbf{C}} \end{bmatrix}\right) \begin{bmatrix} \mathbf{q}_{min}^n \\ \mathbf{q}_{max}^n \end{bmatrix} \quad (3.4.8a)$$

$$\left(\begin{bmatrix} \square & 0 \\ 0 & \square \end{bmatrix} - \frac{\Delta t}{2} \begin{bmatrix} \hat{\mathbf{C}} & 0 \\ 0 & -\hat{\mathbf{C}} \end{bmatrix}\right) \begin{bmatrix} \mathbf{q}_{min}^{n+1} \\ \mathbf{q}_{max}^{n+1} \end{bmatrix} = \left(\begin{bmatrix} \square & 0 \\ 0 & \square \end{bmatrix} + \frac{\Delta t}{2} \begin{bmatrix} \hat{\mathbf{C}} & 0 \\ 0 & -\hat{\mathbf{C}} \end{bmatrix}\right) \begin{bmatrix} \mathbf{q}_{min}^{n+\frac{1}{2}} \\ \mathbf{q}_{max}^{n+\frac{1}{2}} \end{bmatrix} \quad (3.4.8b)$$

where

$$\square = \begin{bmatrix} \square_y & 0 & 0 & 0 & 0 & 0 \\ 0 & \square_z & 0 & 0 & 0 & 0 \\ 0 & 0 & \square_x & 0 & 0 & 0 \\ 0 & 0 & 0 & \square_z & 0 & 0 \\ 0 & 0 & 0 & 0 & \square_x & 0 \\ 0 & 0 & 0 & 0 & 0 & \square_y \end{bmatrix}, \quad \hat{\mathbf{C}} = \frac{1}{2} \begin{bmatrix} \partial_y & 0 & 0 & 0 & 0 & 0 \\ 0 & \partial_z & 0 & 0 & 0 & 0 \\ 0 & 0 & \partial_x & 0 & 0 & 0 \\ 0 & 0 & 0 & \partial_z & 0 & 0 \\ 0 & 0 & 0 & 0 & \partial_x & 0 \\ 0 & 0 & 0 & 0 & 0 & \partial_y \end{bmatrix},$$

$$\mathbf{q}_{min} = \begin{bmatrix} E_x|_{i+\frac{1}{2},\frac{1}{2},k} \\ E_y|_{i,j+\frac{1}{2},\frac{1}{2}} \\ E_z|_{\frac{1}{2},j,k+\frac{1}{2}} \\ E_x|_{i+\frac{1}{2},j,\frac{1}{2}} \\ E_y|_{\frac{1}{2},j+\frac{1}{2},k} \\ E_z|_{i,\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}, \mathbf{q}_{max} = \begin{bmatrix} E_x|_{i+\frac{1}{2},je-\frac{1}{2},k} \\ E_y|_{i,j+\frac{1}{2},ke-\frac{1}{2}} \\ E_z|_{ie-\frac{1}{2},j,k+\frac{1}{2}} \\ E_x|_{i+\frac{1}{2},j,ke-\frac{1}{2}} \\ E_y|_{ie-\frac{1}{2},j+\frac{1}{2},k} \\ E_z|_{i,je-\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}$$

$\square_\xi$ stands for spatial averaging operator along ξ direction, e.g.

$\square_y E_x|_{i+\frac{1}{2},\frac{1}{2},k} \equiv \frac{1}{2}(E_x|_{i+\frac{1}{2},0,k} + E_x|_{i+\frac{1}{2},1,k})$ etc. It can be seen that (3.4.8) conforms to the ADI generalized splitting formulae. With that, we incorporate the Mur ABC into the ADI-FDTD method directly as

$$\left(\hat{\mathbf{D}} - \frac{\Delta t}{2}\hat{\mathbf{A}}_E\right)\mathbf{u}'^{n+\frac{1}{2}} = \left(\hat{\mathbf{D}} + \frac{\Delta t}{2}\hat{\mathbf{B}}_E\right)\mathbf{u}'^n \quad (3.4.9a)$$

$$\left(\hat{\mathbf{D}} - \frac{\Delta t}{2}\hat{\mathbf{B}}_E\right)\mathbf{u}'^{n+1} = \left(\hat{\mathbf{D}} + \frac{\Delta t}{2}\hat{\mathbf{A}}_E\right)\mathbf{u}'^{n+\frac{1}{2}} \quad (3.4.9b)$$

where

$$\mathbf{u}' = \begin{bmatrix} \mathbf{q}_{min} \\ \mathbf{u} \\ \mathbf{q}_{max} \end{bmatrix}, \hat{\mathbf{A}}_E = \begin{bmatrix} \hat{\mathbf{C}} & 0 & 0 \\ 0 & \mathbf{A}_E & 0 \\ 0 & 0 & -\hat{\mathbf{C}} \end{bmatrix}, \hat{\mathbf{B}}_E = \begin{bmatrix} \hat{\mathbf{C}} & 0 & 0 \\ 0 & \mathbf{B}_E & 0 \\ 0 & 0 & -\hat{\mathbf{C}} \end{bmatrix}, \hat{\mathbf{D}} = \begin{bmatrix} \square & 0 & 0 \\ 0 & \mathbf{I}_6 & 0 \\ 0 & 0 & \square \end{bmatrix}$$

Using similar derivation as in Section 3.2.1, we rewrite the original algorithm with Mur ABC in (3.4.9) into the FADI-FDTD scheme as

$$\hat{\mathbf{D}}\mathbf{v}'_E{}^n = \hat{\mathbf{D}}\tilde{\mathbf{u}}'^n - \hat{\mathbf{D}}\mathbf{v}'_E{}^{n-\frac{1}{2}} \quad (3.4.10a)$$

$$\left(\frac{1}{2}\hat{\mathbf{D}} - \frac{\Delta t}{4}\hat{\mathbf{A}}_E\right)\tilde{\mathbf{u}}'^{n+\frac{1}{2}} = \hat{\mathbf{D}}\mathbf{v}'_E{}^n \quad (3.4.10b)$$

$$\hat{\mathbf{D}}\mathbf{v}'_E{}^{n+\frac{1}{2}} = \hat{\mathbf{D}}\tilde{\mathbf{u}}'^{n+\frac{1}{2}} - \hat{\mathbf{D}}\mathbf{v}'_E{}^n \quad (3.4.10c)$$

$$\left(\frac{1}{2}\hat{\mathbf{D}} - \frac{\Delta t}{4}\hat{\mathbf{B}}_E\right)\tilde{\mathbf{u}}'^{n+1} = \hat{\mathbf{D}}\mathbf{v}'_E{}^{n+\frac{1}{2}}. \quad (3.4.10d)$$

with initialization

$$\hat{\mathbf{D}}\mathbf{v}'_E{}^{n-\frac{1}{2}} = \left(\frac{1}{2}\hat{\mathbf{D}} - \frac{\Delta t}{4}\hat{\mathbf{B}}_E\right)\tilde{\mathbf{u}}'^n \quad (3.4.11)$$

where

$$\tilde{\mathbf{u}}' = 2\mathbf{u}', \mathbf{v}'_{\text{E}} = \begin{bmatrix} \mathbf{p}_{\min} \\ \mathbf{v}_{\text{E}} \\ \mathbf{p}_{\max} \end{bmatrix}, \mathbf{p}_{\min} = \begin{bmatrix} e_x|_{i+\frac{1}{2},\frac{1}{2},k} \\ e_y|_{i,j+\frac{1}{2},\frac{1}{2}} \\ e_z|_{\frac{1}{2},j,k+\frac{1}{2}} \\ e_x|_{i+\frac{1}{2},j,\frac{1}{2}} \\ e_y|_{\frac{1}{2},j+\frac{1}{2},k} \\ e_z|_{i,\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}, \mathbf{p}_{\max} = \begin{bmatrix} e_x|_{i+\frac{1}{2},je-\frac{1}{2},k} \\ e_y|_{i,j+\frac{1}{2},ke-\frac{1}{2}} \\ e_z|_{ie-\frac{1}{2},j,k+\frac{1}{2}} \\ e_x|_{i+\frac{1}{2},j,ke-\frac{1}{2}} \\ e_y|_{ie-\frac{1}{2},j+\frac{1}{2},k} \\ e_z|_{i,j,e-\frac{1}{2},k+\frac{1}{2}} \end{bmatrix}$$

The algorithm (3.4.10) is similar to (3.2.10) which constitutes the efficient fundamental scheme that has its RHS free from matrix operators $\hat{\mathbf{A}}_{\text{E}}$ and $\hat{\mathbf{B}}_{\text{E}}$. It reduces the arithmetic operations which leads to computationally efficient updating equations. This reduces the flops count which in turn increases the overall efficiency.

Novel Implementation

We now look into the formulation of the novel implementation for the FADI-FDTD method. By applying central difference approximation in (3.4.2), we obtain

For first procedure from n to $n + 1/2$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},\frac{1}{2},k}^n \quad (3.4.12a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+1} = E_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}}. \quad (3.4.12b)$$

As the RHS of (3.4.12) does not involve any operators, it cannot be conformed to the ADI algorithm. To incorporate the Mur ABC into the FADI-FDTD method, we will first apply spatial averaging on the RHS of (3.4.12) followed by re-definition of field variables in (3.2.8) to obtain the following:

For first procedure from n to $n + 1/2$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) \tilde{E}_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{1}{2} \left(\tilde{E}_x|_{i+\frac{1}{2},0,k}^n + \tilde{E}_x|_{i+\frac{1}{2},1,k}^n \right) \quad (3.4.13a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) \tilde{E}_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+1} = \frac{1}{2} \left(\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \right). \quad (3.4.13b)$$

By substituting (3.2.12a) and (3.2.15a) into (3.4.13a) and (3.4.13b) respectively, we get

For first procedure from n to $n + 1/2$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) \tilde{E}_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{1}{2} \left(\tilde{E}_x|_{i+\frac{1}{2},0,k}^n + e_x|_{i+\frac{1}{2},1,k}^n + e_x|_{i+\frac{1}{2},1,k}^{n-\frac{1}{2}} \right) \quad (3.4.14a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) \tilde{E}_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+1} = \frac{1}{2} \left(\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^n \right). \quad (3.4.14b)$$

Note that it is not required for $\tilde{E}_x|_{i+\frac{1}{2},0,k}$ in (3.4.14) to be converted into a auxiliary variable. This is due to $\tilde{E}_x|_{i+\frac{1}{2},0,k}$ being a boundary variable and is not involved for the remaining update equations.

With the formulation of the Mur ABC for both consistent and novel implementations in 3-D FADI-FDTD method, the updating equations for $\tilde{E}_x$ variable at boundaries $y = 0$ and $z = 0$ are discretized as follows:

i) Consistent Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y}\right) \tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y}\right) \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2},0,k}^n + e_x|_{i+\frac{1}{2},1,k}^n \quad (3.4.15a)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(e_x|_{i+\frac{1}{2},j,0}^n + e_x|_{i+\frac{1}{2},j,1}^n \right) \quad (3.4.15b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+1} = e_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (3.4.16a)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \right) \quad (3.4.16b)$$

ii) Novel Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta y} \right) \tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta y} \right) \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},0,k}^n + e_x|_{i+\frac{1}{2},1,k}^n + e_x|_{i+\frac{1}{2},1,k}^{n-\frac{1}{2}} \quad (3.4.17a)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{\Delta z}{\Delta z + v\Delta t} \left(\tilde{E}_x|_{i+\frac{1}{2},j,0}^n + e_x|_{i+\frac{1}{2},j,1}^n + e_x|_{i+\frac{1}{2},j,1}^{n-\frac{1}{2}} \right) \quad (3.4.17b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+1} = \tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,1}^n \quad (3.4.18a)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{\Delta y}{\Delta y + v\Delta t} \left(\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^n \right) \quad (3.4.18b)$$

The complete update equations for consistent and novel implementations of the Mur ABC in 3-D FADI-FDTD method are shown in Appendix D. It can be seen that some field variables on the RHS of (3.4.3)-(3.4.6) have been changed to the expressions in terms of auxiliary variables for the efficient fundamental scheme in (3.4.15)-(3.4.18). Furthermore, by comparing (3.4.17)-(3.4.18) with (3.4.15)-(3.4.16), there is an extra variable on the RHS of (3.4.17)-(3.4.18). Such overhead will be examined through the CPU time of both consistent and novel implementations in Section 3.4.4.

3.4.3 Mur ABC for FLOD-FDTD Method

Similar to the incorporation of the Mur ABC into the FADI-FDTD, we present the consistent and novel implementations of the Mur ABC into the FLOD-FDTD method. It is relatively easier for both consistent and novel implementations of the Mur ABC to be incorporated into the FLOD-FDTD method as compared to the FADI-FDTD method. The consistent and novel implementations of the Mur ABC can be incorporated into the FLOD-FDTD method by substituting the auxiliary variables of the FLOD-FDTD method in (3.2.24) into the Mur ABC update equations as shown below.

Consistent Implementation

With reference to the consistent implementation of the Mur ABC in (3.4.7), by substituting (3.2.27a) and (3.2.31a) into (3.4.7a) and (3.4.7b) respectively, we arrive at

For first procedure from n to $n + 1/2$:

$$\left(1 - \frac{v\Delta t}{4} \frac{\partial}{\partial y}\right) e_x|_{i+\frac{1}{2},\frac{1}{2},k}^{n+\frac{1}{2}} = 2E_x|_{i+\frac{1}{2},\frac{1}{2},k}^n \quad (3.4.19a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\left(1 - \frac{v\Delta t}{4} \frac{\partial}{\partial y}\right) e_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+1} = 2E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{2}}. \quad (3.4.19b)$$

Novel Implementation

By performing similar substitution of (3.2.27a) and (3.2.31a) into (3.4.12a) and (3.4.12b) of the novel implementation respectively, we have

For first procedure from n to $n + 1/2$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) e_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{2}} = \left(2 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^n \quad (3.4.20a)$$

For second procedure from $n + 1/2$ to $n + 1$:

$$\left(1 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) e_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+1} = \left(2 - \frac{v\Delta t}{2} \frac{\partial}{\partial y}\right) E_x|_{i+\frac{1}{2}, \frac{1}{2}, k}^{n+\frac{1}{2}}. \quad (3.4.20b)$$

It can be seen that the differential operators have been reintroduced into the RHS of the novel implementation in (3.4.20). This is due to the formulation of the FLOD-FDTD method which does not require any initialization and the resultant field solution can be obtained directly. Note that only the update equations on the boundaries for the FLOD-FDTD method requires differential operators. The interior computation domain of the FLOD-FDTD method remains RHS matrix-operators free, therefore the overall efficiency still improves (to be shown in Table 3.6). Furthermore, by implementing the novel implementation, a lower reflection coefficient can be achieved as compared to the consistent implementation (to be shown in Figure 3.4).

For 3-D FLOD-FDTD method, the update equations for e_x variable at boundaries $y = 0$ and $z = 0$ of the Mur ABC are as follows:

i) Consistent Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y}\right)e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y}\right)e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},0,k}^n + E_x|_{i+\frac{1}{2},1,k}^n \quad (3.4.21a)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t}e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,0}^n + E_x|_{i+\frac{1}{2},j,1}^n\right) \quad (3.4.21b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z}\right)e_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z}\right)e_x|_{i+\frac{1}{2},j,1}^{n+1} = E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (3.4.22a)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t}e_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}}\right) \quad (3.4.22b)$$

ii) Novel Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta y}\right)e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta y}\right)e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta y}\right)E_x|_{i+\frac{1}{2},0,k}^n + \left(1 - \frac{v\Delta t}{2\Delta y}\right)E_x|_{i+\frac{1}{2},1,k}^n \end{aligned} \quad (3.4.23a)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t}e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{2\Delta z + v\Delta t}{\Delta z + v\Delta t}E_x|_{i+\frac{1}{2},j,0}^n + \frac{2\Delta z - v\Delta t}{\Delta z + v\Delta t}E_x|_{i+\frac{1}{2},j,1}^n \quad (3.4.23b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{2\Delta z}\right)e_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta z}\right)e_x|_{i+\frac{1}{2},j,1}^{n+1}$$

$$= \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (3.4.24a)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} e_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{2\Delta y + v\Delta t}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \frac{2\Delta y - v\Delta t}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \quad (3.4.24b)$$

The complete update equations of the Mur ABC for both consistent and novel implementations in 3-D FLOD-FDTD method are in Appendix E. It can be seen that the LHS of (3.4.3)-(3.4.6) has been changed to the expressions in terms of auxiliary variables for the efficient fundamental scheme in (3.4.21)-(3.4.24). Meanwhile, although there are extra update coefficients on the RHS of (3.4.23)-(3.4.24) as compared to (3.4.21)-(3.4.22), such overhead is considered negligible as to be examined below through the CPU time of both implementations.

3.4.4 Numerical Results

In this section, both Mur ABC implementations in FADI- and FLOD-FDTD methods are compared and validated with the conventional counterparts through numerical simulations. A line electric current source is applied at the center of a free space domain with $650 \times 650 \times 5$ grids and spatial step $\Delta x = \Delta y = \Delta z = 1.0$ mm. A sinusoid modulated Gaussian pulse is utilized as excitation:

$$J_z = e^{-\left(\frac{t-t_0}{\tau}\right)^2} \sin\left(2\pi f_0(t-t_0)\right) \quad (3.4.25)$$

where $f_0 = 15$ GHz, $\tau = 60$ ps, $t_0 = 4\tau$.

The reflection coefficient for the Mur ABC is studied with the observation point located twenty cells away from the source and one cell away from the Mur ABC

Figure 3.3: The reflection coefficients of the Mur ABC for (a) consistent implementation and (b) novel implementation in the conventional and efficient fundamental 3-D ADI-FDTD methods with various CFLN.

Figure 3.4: The reflection coefficients of the Mur ABC for (a) consistent implementation and (b) novel implementation in the conventional and efficient fundamental 3-D LOD-FDTD methods with various CFLN.

interface. The reflection coefficient for the Mur ABC is evaluated by

$$R_C = 20 \log_{10} \left| \frac{FT\{E - E_{ref}\}}{FT\{E_{ref}\}} \right| \quad (3.4.26)$$

where FT refers to Fourier transform and E_{ref} is the reference value calculated in a grid large enough so that any reflection from the boundary is isolated.

Figures 3.3 and 3.4 show the reflection coefficients of the Mur ABC for consistent and novel implementations in the conventional and efficient fundamental 3-D ADI- and LOD-FDTD methods with various $CFLN = \Delta t / \Delta t_{CFL}$, where Δt_{CFL} is the CFL limit in the Yee's explicit FDTD method. It demonstrates that the numerical results of both conventional and efficient fundamental schemes are identical, and validates that both methods are equivalent, but with greater efficiency and simplicity for the latter. It is worth noting that the reflection coefficients for consistent and novel implementations are similar in both ADI- and LOD-FDTD methods. This shows the effectiveness of the Mur ABC for these two implementations. Between these two implementations, it can be seen that the novel implementation has lower reflection coefficient than the consistent implementation. For comparison, we have added the reflection coefficient for the Mur ABC in the Yee's explicit FDTD. Although they may degrade for large CFLN, the reflection coefficients for ADI- and LOD-FDTD methods (with novel implementation) are comparable or lower than those for the Yee's explicit FDTD method.

For the novel implementation of the Mur ABC, the reflected waves in the frequency domain are skewed towards the higher frequency region. The skewing gets greater with $CFLN \geq 1$. This causes the lower frequency region to have a lower reflection, leading to a better performance of the reflection coefficient with $CFLN \geq 1$. However, this is not the case for the consistent implementation. The reflected waves of the consistent implementation do not skew. Therefore, the reflection coefficient

Table 3.4: Flops count of consistent and novel implementations for the Mur ABC

Scheme		ADI/LOD-FDTD	FADI-FDTD	FLOD-FDTD
Consistent Mur ABC (3.4.1) (3.4.7) & (3.4.19)	A/S	3	3	3
	M/D	3	2	2
	Total	6	5	5
Novel Mur ABC (3.4.2) (3.4.12) & (3.4.20)	A/S	3	5	3
	M/D	2	2	5
	Total	5	7	8

deteriorates as the CFLN increases at low frequencies. On the other hand, at higher frequencies, the wavelength of the waveform for both consistent and novel implementations will get shorter, causing the sampling to decrease. This in turn results in the performance of the reflection coefficient to deteriorate with $\text{CFLN} \geq 1$. To compare the performance of both consistent and novel implementations, we consider the overall (average) reflection coefficient of the frequency range from 8 to 26 GHz (to be shown in Tables 3.5 and 3.6).

Next, we compare the flops count of consistent and novel implementations for the Mur ABC in the conventional and efficient fundamental schemes shown in Table 3.4. The comparison of these Mur ABC update equations are based on their respective RHS. From Table 3.4, it can be seen that the total flops count (A/S+M/D) has been reduced for the Mur ABC with consistent implementation using the efficient fundamental schemes. However, the total flops count increases for the Mur ABC with novel implementation using the efficient fundamental schemes. To investigate the effects of the flops count with efficiency gain, we further compare the CPU time of both conventional and efficient fundamental ADI- and LOD-FDTD methods. The numerical simulation is performed for various CFLN with the computing platform of Microsoft Visual C++ environment with Intel Dual Core 2.66 GHz processor.

Table 3.5: CPU time and average reflection coefficient for (a) consistent implementation and (b) novel implementation of the conventional and efficient fundamental ADI schemes

Mur ABC	Consistent Implementation			
CFLN	1	2	4	8
ADI	8481.6s	4183.5s	2116.1s	1055.1s
FADI	4791.2s	2377.0s	1185.5s	596.1s
Eff. Gain	1.77	1.76	1.78	1.77
Avg. R_C	-26.90	-23.44	-16.79	-10.30

(a) Consistent implementation

Mur ABC	Novel Implementation			
CFLN	1	2	4	8
ADI	8479.2s	4185.9s	2115.4s	1054.7s
FADI	4958.6s	2476.9s	1251.7s	616.8s
Eff. Gain	1.71	1.69	1.69	1.71
Avg. R_C	-32.09	-31.40	-28.53	-22.80

(b) Novel implementation

Tables 3.5 and 3.6 show the CPU time and average reflection coefficient of the conventional and efficient fundamental ADI and LOD methods. We find that the efficiency gains for various CFLNs in both implementations of the Mur ABC in the FLOD-FDTD method are close (1.45-1.47), which confirms that the overhead on the RHS of (3.4.23)-(3.4.24) for the novel implementation is negligible. On the other hand, the efficiency gain for various CFLNs in the novel implementation of the Mur ABC in the FADI-FDTD method is slightly lower as compared to the consistent implementation. This is due to an extra variable required on the RHS of (3.4.17)-(3.4.18) for the novel implementation. Furthermore, it can be seen that as the CFLN increases, the average reflection coefficient for both implementations of the Mur ABC in ADI- and LOD-FDTD methods generally increases. This shows a

Table 3.6: CPU time and average reflection coefficient for (a) consistent implementation and (b) novel implementation of the conventional and efficient fundamental LOD schemes

Mur ABC	Consistent Implementation			
CFLN	1	2	4	8
LOD	6968.7s	3483.6s	1729.0.8s	875.4s
FLOD	4773.1s	2386.2s	1192.4s	595.5s
Eff. Gain	1.46	1.46	1.45	1.47
Avg. R_C	-26.75	-23.25	-16.67	-10.29

(a) Consistent implementation

Mur ABC	Novel Implementation			
CFLN	1	2	4	8
LOD	6966.1s	3457.7s	1730.1s	874.8s
FLOD	4771.3s	2384.6s	1193.2s	595.1s
Eff. Gain	1.46	1.45	1.45	1.47
Avg. R_C	-31.92	-31.26	-28.51	-22.95

(b) Novel implementation

trade-off between accuracy and efficiency for both ADI- and LOD-FDTD methods.

3.5 Split-Field PML for ADI-FDTD Method

In this section, we discuss the formulation of the split-field PML for 2-D TE FADI-FDTD method. Numerical simulations in terms of reflection error for both methods are performed for comparison. The efficiency gain for the FADI-FDTD method over conventional ADI-FDTD method with split-field PML will be presented.

3.5.1 ADI-FDTD method with Split-field PML

Using the conventional 2-D TE ADI splitting formulae, the split-field formulation can be written as

$$\left(\mathbf{I}_4 - \frac{\Delta t}{2}(\mathbf{A}_E + \frac{\hat{\mathbf{L}}}{2})\right)\mathbf{u}^{n+\frac{1}{2}} = \left(\mathbf{I}_4 + \frac{\Delta t}{2}(\mathbf{B}_E + \frac{\hat{\mathbf{L}}}{2})\right)\mathbf{u}^n \quad (3.5.1a)$$

$$\left(\mathbf{I}_4 - \frac{\Delta t}{2}(\mathbf{B}_E + \frac{\hat{\mathbf{L}}}{2})\right)\mathbf{u}^{n+1} = \left(\mathbf{I}_4 + \frac{\Delta t}{2}(\mathbf{A}_E + \frac{\hat{\mathbf{L}}}{2})\right)\mathbf{u}^{n+\frac{1}{2}} \quad (3.5.1b)$$

where

$$\mathbf{u} = \begin{bmatrix} E_x \\ E_y \\ H_{zx} \\ H_{zy} \end{bmatrix}, \mathbf{A}_E = \begin{bmatrix} 0 & 0 & \frac{\partial_y}{\epsilon} & \frac{\partial_y}{\epsilon} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\partial_y}{\mu} & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-\partial_x}{\epsilon} & \frac{-\partial_x}{\epsilon} \\ 0 & \frac{-\partial_x}{\mu} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\hat{\mathbf{L}} = \begin{bmatrix} \frac{-\sigma_y}{\epsilon} & 0 & 0 & 0 \\ 0 & \frac{-\sigma_x}{\epsilon} & 0 & 0 \\ 0 & 0 & \frac{-\sigma_x^m}{\mu} & 0 \\ 0 & 0 & 0 & \frac{-\sigma_y^m}{\mu} \end{bmatrix}$$

and $\mathbf{I}_4$ is an 4×4 identity matrix. Note that variables $\mathbf{u}$, $\mathbf{A}_E$ and $\mathbf{B}_E$ are for 2-D TE ADI-FDTD method and they are different from (2.1.7), (2.1.8) and (2.1.9) respectively.

For first procedure from n to $n + \frac{1}{2}$, we have

$$\begin{aligned} E_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} &= \frac{\alpha_{y_{i+\frac{1}{2},j}}^\epsilon}{\beta_{y_{i+\frac{1}{2},j}}^\epsilon} E_x|_{i+\frac{1}{2},j}^n \\ &+ \frac{a_{1,y}}{\beta_{y_{i+\frac{1}{2},j}}^\epsilon} \left[H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - H_{zx}|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} + H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - H_{zy}|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} \right] \end{aligned} \quad (3.5.2a)$$

$$E_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{\alpha_{x_{i,j+\frac{1}{2}}}^\epsilon}{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon} E_y|_{i,j+\frac{1}{2}}^n$$

$$-\frac{a_{1,x}}{\beta_{x,i,j+\frac{1}{2}}^\epsilon} \left[H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - H_{zx}|_{i-\frac{1}{2},j+\frac{1}{2}}^n + H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - H_{zy}|_{i-\frac{1}{2},j+\frac{1}{2}}^n \right] \quad (3.5.2b)$$

$$H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{\alpha_{x,i+\frac{1}{2},j+\frac{1}{2}}^\mu}{\beta_{x,i+\frac{1}{2},j+\frac{1}{2}}^\mu} H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - \frac{a_{2,x}}{\beta_{x,i+\frac{1}{2},j+\frac{1}{2}}^\mu} \left[E_y|_{i+1,j+\frac{1}{2}}^n - E_y|_{i,j+\frac{1}{2}}^n \right] \quad (3.5.2c)$$

$$H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{\alpha_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu}{\beta_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu} H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n + \frac{a_{2,y}}{\beta_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu} \left[E_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right] \quad (3.5.2d)$$

where

$$a_1 = \frac{\Delta t}{2\epsilon}, \quad a_2 = \frac{\Delta t}{2\mu}, \quad a_{1,\tilde{\xi}} = \frac{\Delta t}{2\epsilon\Delta\tilde{\xi}}, \quad a_{2,\tilde{\xi}} = \frac{\Delta t}{2\mu\Delta\tilde{\xi}},$$

$$\alpha_{\tilde{\xi}}^\epsilon = \left(1 - a_1 \frac{\sigma_{\tilde{\xi}}}{2}\right), \quad \beta_{\tilde{\xi}}^\epsilon = \left(1 + a_1 \frac{\sigma_{\tilde{\xi}}}{2}\right), \quad \alpha_{\tilde{\xi}}^\mu = \left(1 - a_2 \frac{\sigma_{\tilde{\xi}}^m}{2}\right), \quad \beta_{\tilde{\xi}}^\mu = \left(1 + a_2 \frac{\sigma_{\tilde{\xi}}^m}{2}\right).$$

By substituting (3.5.2c) and (3.5.2d) into (3.5.2a), we arrive at a tridiagonal matrix for E_x as

$$\begin{aligned} & \frac{-a_{1,y}a_{2,y}}{\beta_{y,i+\frac{1}{2},j-\frac{1}{2}}^\mu} E_x|_{i+\frac{1}{2},j-1}^{n+\frac{1}{2}} + \widehat{\gamma}_y E_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - \frac{a_{1,y}a_{2,y}}{\beta_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu} E_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} \\ &= \alpha_{y,i+\frac{1}{2},j}^\epsilon E_x|_{i+\frac{1}{2},j}^n + a_{1,y} \left[\frac{\alpha_{x,i+\frac{1}{2},j+\frac{1}{2}}^\mu}{\beta_{x,i+\frac{1}{2},j+\frac{1}{2}}^\mu} H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - \frac{\alpha_{x,i+\frac{1}{2},j-\frac{1}{2}}^\mu}{\beta_{x,i+\frac{1}{2},j-\frac{1}{2}}^\mu} H_{zx}|_{i+\frac{1}{2},j-\frac{1}{2}}^n \right. \\ & \quad \left. + \frac{\alpha_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu}{\beta_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu} H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - \frac{\alpha_{y,i+\frac{1}{2},j-\frac{1}{2}}^\mu}{\beta_{y,i+\frac{1}{2},j-\frac{1}{2}}^\mu} H_{zy}|_{i+\frac{1}{2},j-\frac{1}{2}}^n \right] \\ & \quad - \frac{a_{1,y}a_{2,x}}{\beta_{x,i+\frac{1}{2},j+\frac{1}{2}}^\mu} \left[E_y|_{i+1,j+\frac{1}{2}}^n - E_y|_{i,j+\frac{1}{2}}^n \right] + \frac{a_{1,y}a_{2,x}}{\beta_{x,i+\frac{1}{2},j-\frac{1}{2}}^\mu} \left[E_y|_{i+1,j-\frac{1}{2}}^n - E_y|_{i,j-\frac{1}{2}}^n \right] \end{aligned} \quad (3.5.3)$$

with

$$\widehat{\gamma}_y = \beta_{y,i+\frac{1}{2},j}^\epsilon + \frac{a_{1,y}a_{2,y}}{\beta_{y,i+\frac{1}{2},j-\frac{1}{2}}^\mu} + \frac{a_{1,y}a_{2,y}}{\beta_{y,i+\frac{1}{2},j+\frac{1}{2}}^\mu}.$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$, we get

$$E_x|_{i+\frac{1}{2},j}^{n+1} = \frac{\alpha_{y,i+\frac{1}{2},j}^\epsilon}{\beta_{y,i+\frac{1}{2},j}^\epsilon} E_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$$

$$+ \frac{a_{1,y}}{\beta_{y_{i+\frac{1}{2},j}}^\epsilon} \left[H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - H_{zx}|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} + H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - H_{zy}|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} \right] \quad (3.5.4a)$$

$$\begin{aligned} E_y|_{i,j+\frac{1}{2}}^{n+1} &= \frac{\alpha_{x_{i,j+\frac{1}{2}}}^\epsilon}{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon} E_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \\ &- \frac{a_{1,x}}{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon} \left[H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - H_{zx}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+1} + H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - H_{zy}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+1} \right] \end{aligned} \quad (3.5.4b)$$

$$H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \frac{\alpha_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{2,x}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \left[E_y|_{i+1,j+\frac{1}{2}}^{n+1} - E_y|_{i,j+\frac{1}{2}}^{n+1} \right] \quad (3.5.4c)$$

$$H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \frac{\alpha_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{a_{2,y}}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \left[E_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right] \quad (3.5.4d)$$

Using a similar approach, by substituting (3.5.4c) and (3.5.4d) into (3.5.4b), we obtain the tridiagonal matrix for E_y as

$$\begin{aligned} &\frac{-a_{1,x}a_{2,x}}{\beta_{x_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} E_y|_{i-1,j+\frac{1}{2}}^{n+1} + \hat{\gamma}_x E_y|_{i,j+\frac{1}{2}}^{n+1} - \frac{a_{1,x}a_{2,x}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} E_y|_{i+1,j+\frac{1}{2}}^{n+1} \\ &= \alpha_{x_{i,j+\frac{1}{2}}}^\epsilon E_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - a_{1,x} \left[\frac{\alpha_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} H_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\alpha_{x_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu}{\beta_{x_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} H_{zx}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \right. \\ &\quad \left. + \frac{\alpha_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} H_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{\alpha_{y_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu}{\beta_{y_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} H_{zy}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \right] \\ &\quad - \frac{a_{1,x}a_{2,y}}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \left[E_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right] + \frac{a_{1,x}a_{2,y}}{\beta_{y_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} \left[E_x|_{i-\frac{1}{2},j+1}^{n+\frac{1}{2}} - E_x|_{i-\frac{1}{2},j}^{n+\frac{1}{2}} \right] \end{aligned} \quad (3.5.5)$$

with

$$\hat{\gamma}_x = \beta_{x_{i,j+\frac{1}{2}}}^\epsilon + \frac{a_{1,x}a_{2,x}}{\beta_{x_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} + \frac{a_{1,x}a_{2,x}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}.$$

3.5.2 FADI-FDTD Method with Split-Field PML

The previous update equations for the conventional ADI-FDTD method with split-field PML require many arithmetic operations for their RHS. To maximize the

efficiency, we modify the conventional ADI-FDTD method with split-field PML in (3.5.1) as

$$\mathbf{v}_E^n = \tilde{\mathbf{u}}^n - \mathbf{v}_E^{n-\frac{1}{2}} \quad (3.5.6a)$$

$$\left(\frac{1}{2}\mathbf{I}_4 - \frac{\Delta t}{4}(\mathbf{A}_E + \frac{\hat{\mathbf{L}}}{2})\right)\tilde{\mathbf{u}}^{n+\frac{1}{2}} = \mathbf{v}_E^n \quad (3.5.6b)$$

$$\mathbf{v}_E^{n+\frac{1}{2}} = \tilde{\mathbf{u}}^{n+\frac{1}{2}} - \mathbf{v}_E^n \quad (3.5.6c)$$

$$\left(\frac{1}{2}\mathbf{I}_4 - \frac{\Delta t}{4}(\mathbf{B}_E + \frac{\hat{\mathbf{L}}}{2})\right)\tilde{\mathbf{u}}^{n+1} = \mathbf{v}_E^{n+\frac{1}{2}} \quad (3.5.6d)$$

with initialization

$$\mathbf{v}_E^{-\frac{1}{2}} = \left(\frac{1}{2}\mathbf{I}_4 - \frac{\Delta t}{4}(\mathbf{B}_E + \frac{\hat{\mathbf{L}}}{2})\right)\tilde{\mathbf{u}}^n \quad (3.5.7)$$

where

$$\begin{aligned} \mathbf{v}_E &= [e_x, e_y, h_{zx}, h_{zy}]^T, \quad \tilde{\mathbf{u}} = [\tilde{E}_x, \tilde{E}_y, \tilde{H}_{zx}, \tilde{H}_{zy}]^T, \\ \tilde{\mathbf{u}}^n &= 2\mathbf{u}^n, \quad \tilde{\mathbf{u}}^{n+\frac{1}{2}} = 2\mathbf{u}^{n+\frac{1}{2}}, \quad \tilde{\mathbf{u}}^{n+1} = 2\mathbf{u}^{n+1}. \end{aligned}$$

The $\mathbf{v}_E$'s serve as auxiliary variables which do not require additional memory. Note that the auxiliary variable $\mathbf{v}_E$ is for 2-D TE FADI-FDTD method which is different from (3.2.2).

For first procedure from n to $n + \frac{1}{2}$, we now have from (3.5.6a)-(3.5.6b),

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j}^n = \tilde{E}_x|_{i+\frac{1}{2},j}^n - e_x|_{i+\frac{1}{2},j}^{n-\frac{1}{2}} \quad (3.5.8a)$$

$$e_y|_{i,j+\frac{1}{2}}^n = \tilde{E}_y|_{i,j+\frac{1}{2}}^n - e_y|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} \quad (3.5.8b)$$

$$h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n = \tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n-\frac{1}{2}} \quad (3.5.8c)$$

$$h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n = \tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n-\frac{1}{2}}. \quad (3.5.8d)$$

b) Implicit update for $\tilde{E}$ and explicit update for $\tilde{H}$

$$\begin{aligned}\tilde{E}_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} &= \frac{2}{\beta_{y_{i+\frac{1}{2},j}}^\epsilon} e_x|_{i+\frac{1}{2},j}^n \\ &+ \frac{a_{1,y}}{\beta_{y_{i+\frac{1}{2},j}}^\epsilon} \left[\tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_{zx}|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} + \tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_{zy}|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} \right]\end{aligned}\quad (3.5.9a)$$

$$\tilde{E}_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{2}{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon} e_y|_{i,j+\frac{1}{2}}^n \quad (3.5.9b)$$

$$\tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{2}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n \quad (3.5.9c)$$

$$\tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{2}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n + \frac{a_{2,y}}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \left[\tilde{E}_x|_{i+\frac{1}{2},j+1}^{n+\frac{1}{2}} - \tilde{E}_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \right]. \quad (3.5.9d)$$

By substituting (3.5.9c) and (3.5.9d) into (3.5.9a), a tridiagonal matrix for $\tilde{E}_x$ is obtained as

$$\begin{aligned}\frac{1}{2} \left[\frac{-a_{1,y}a_{2,y}}{\beta_{y_{i+\frac{1}{2},j-\frac{1}{2}}}^\mu} \tilde{E}_x|_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}} + \hat{\gamma}_y \tilde{E}_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - \frac{a_{1,y}a_{2,y}}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \tilde{E}_x|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \right] \\ = e_x|_{i+\frac{1}{2},j}^n + \left[\frac{a_{1,y}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - \frac{a_{1,y}}{\beta_{x_{i+\frac{1}{2},j-\frac{1}{2}}}^\mu} h_{zx}|_{i+\frac{1}{2},j-\frac{1}{2}}^n \right. \\ \left. + \frac{a_{1,y}}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n - \frac{a_{1,y}}{\beta_{y_{i+\frac{1}{2},j-\frac{1}{2}}}^\mu} h_{zy}|_{i+\frac{1}{2},j-\frac{1}{2}}^n \right]. \quad (3.5.10)\end{aligned}$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$, we can obtain from (3.5.6c)-(3.5.6d),

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - e_x|_{i+\frac{1}{2},j}^n \quad (3.5.11a)$$

$$e_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{E}_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - e_y|_{i,j+\frac{1}{2}}^n \quad (3.5.11b)$$

$$h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^n \quad (3.5.11c)$$

$$h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^n. \quad (3.5.11d)$$

b) Implicit update for $\tilde{E}$ and explicit update for $\tilde{H}$

$$\tilde{E}_x|_{i+\frac{1}{2},j}^{n+1} = \frac{2}{\beta_{y_{i+\frac{1}{2},j}}^\epsilon} e_x|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \quad (3.5.12a)$$

$$\begin{aligned} \tilde{E}_y|_{i,j+\frac{1}{2}}^{n+1} &= \frac{2}{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon} e_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \\ &\quad - \frac{a_{1,x}}{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon} \left[\tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - \tilde{H}_{zx}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+1} + \tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} - \tilde{H}_{zy}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+1} \right] \end{aligned} \quad (3.5.12b)$$

$$\tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \frac{2}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{2,x}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \left[\tilde{E}_y|_{i+1,j+\frac{1}{2}}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2}}^{n+1} \right] \quad (3.5.12c)$$

$$\tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} = \frac{2}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}}. \quad (3.5.12d)$$

By substituting (3.5.12c) and (3.5.12d) into (3.5.12b), a tridiagonal matrix for $\tilde{E}_y$ is obtained as

$$\begin{aligned} \frac{1}{2} \left[\frac{-a_{1,x}a_{2,x}}{\beta_{x_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} \tilde{E}_y|_{i-1,j+\frac{1}{2}}^{n+1} + \hat{\gamma}_x \tilde{E}_y|_{i,j+\frac{1}{2}}^{n+1} - \frac{a_{1,x}a_{2,x}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} \tilde{E}_y|_{i+1,j+\frac{1}{2}}^{n+1} \right] \\ = e_y|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - \left[\frac{a_{1,x}}{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{1,x}}{\beta_{x_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zx}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \right. \\ \left. + \frac{a_{1,x}}{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{1,x}}{\beta_{y_{i-\frac{1}{2},j+\frac{1}{2}}}^\mu} h_{zy}|_{i-\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} \right]. \end{aligned} \quad (3.5.13)$$

For non-zero initial $\tilde{E}^0$ and $\tilde{H}^0$, we apply the initialization as follows:

$$e_x|_{i+\frac{1}{2},j}^{-\frac{1}{2}} = \frac{\beta_{y_{i+\frac{1}{2},j}}^\epsilon}{2} \tilde{E}_x|_{i+\frac{1}{2},j}^0 \quad (3.5.14a)$$

$$\begin{aligned} e_y|_{i,j+\frac{1}{2}}^{-\frac{1}{2}} &= \frac{\beta_{x_{i,j+\frac{1}{2}}}^\epsilon}{2} \tilde{E}_y|_{i,j+\frac{1}{2}}^0 \\ &\quad + \frac{a_{1,x}}{2} \left[\tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^0 - \tilde{H}_{zx}|_{i-\frac{1}{2},j+\frac{1}{2}}^0 + \tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^0 - \tilde{H}_{zy}|_{i-\frac{1}{2},j+\frac{1}{2}}^0 \right] \end{aligned} \quad (3.5.14b)$$

$$h_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^{-\frac{1}{2}} = \frac{\beta_{x_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}{2} \tilde{H}_{zx}|_{i+\frac{1}{2},j+\frac{1}{2}}^0 + \frac{a_{2,x}}{2} \left[\tilde{E}_y|_{i+1,j+\frac{1}{2}}^0 - \tilde{E}_y|_{i,j+\frac{1}{2}}^0 \right] \quad (3.5.14c)$$

$$h_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^{-\frac{1}{2}} = \frac{\beta_{y_{i+\frac{1}{2},j+\frac{1}{2}}}^\mu}{2} \tilde{H}_{zy}|_{i+\frac{1}{2},j+\frac{1}{2}}^0. \quad (3.5.14d)$$

Upon comparison between both procedures of the conventional and efficient fundamental ADI-FDTD, we find that the update equations are more concise and simplified for the latter. Furthermore, there is a decrease in arithmetic operations. With the overall flops count decreases, the overall efficiency is increased.

3.5.3 Numerical Results

We conduct numerical simulations for both conventional and efficient fundamental ADI-FDTD methods with split-field PML incorporated. The computation domain with spatial step $\Delta x = \Delta y = 1.33$ mm and lattice dimension (including the PML region) of 42×42 grids are used. Ten layers of PML are used in both x - and y -directions with a conductivity profile [39] given in (2.1.36) and a choice for $\sigma_{\xi_{max}}$ that will minimize reflection is expressed in (2.1.38). In our study, the split-field PML parameters are $\sigma_{max} = \sigma_{opt}$, $m = 4$ and $\Delta t_{CFL} = 3.137$ ps.

A point source excitation is located at the center of the computation region. The source is a differentiated Gaussian pulse applied to H_z component as

$$M_z = -2 \frac{(t - t_0)}{\tau} e^{-\left(\frac{t-t_0}{\tau}\right)^2}, \tau = 82 \text{ ps}, t_0 = 3\tau. \quad (3.5.15)$$

The reflection error for the split-field PML is studied with the observation point located ten cells away from the source and one cell away from the PML interface. The reflection error for the split-field PML is evaluated by

$$R_H = 20 \log_{10} \left(\frac{|H^n - H_{ref}^n|}{\max |H_{ref}^n|} \right) \quad (3.5.16)$$

where H_{ref}^n is the reference value calculated in a grid large enough so that any reflection from the boundary is isolated.

Figure 3.5 plots the reflection errors for both conventional and efficient fundamental ADI-FDTD methods with split-field PML incorporated for various CFLN. We

Figure 3.5: Reflection errors for the conventional and efficient fundamental ADI-FDTD methods with split-field PML incorporated for various CFLN.

observe that the numerical results are identical which confirm that both implementations are equivalently effective. In addition, with increasing CFLN, the reflection error for the split-field PML deteriorates. This shows a good trade-off between accuracy and efficiency of the ADI-FDTD method with split-field PML. As the FADI-FDTD method provides computationally efficient update equations, it is therefore more appealing than the conventional implementation.

Next, we compare the flops count for the split-field PML using conventional and efficient fundamental ADI-FDTD methods shown in Table 3.7. The comparison of these update equations (inclusive of split-field PML) in the conventional and efficient fundamental schemes is based on their respective RHS. It can be seen that the number of flops decreases for the FADI-FDTD method with split-field PML as compare to its conventional counterpart. To justify the efficiency gain of the FADI-FDTD method with split-field PML, a numerical simulation is performed for 1000

Table 3.7: Flops count for the split-field PML using conventional and efficient fundamental ADI-FDTD methods

Split-Field PML		ADI-FDTD (3.5.1)	FADI-FDTD (3.5.6)
Implicit	A/S	16	8
	M/D	14	8
Explicit	A/S	16	12
	M/D	12	8
Total	A/S	32	20
	M/D	26	16
	A/S + M/D	58	36

time steps at $CFLN = 4$ in a computation domain of 500×500 grids. The computing platform is Microsoft Visual C++ environment with Intel Dual Core 2.66 GHz processor. We find that the efficiency gain achieved by the efficient fundamental scheme over conventional scheme is approximately 1.44. It is thus ascertained that substantial gain in the overall efficiency is achievable for the FADI-FDTD method even with split-field PML incorporated.

3.6 CFS-CPML for ADI-FDTD Method

The derivation of 3-D FADI-FDTD method with CFS-CPML will be demonstrated. It is then incorporated into the GPU using CUDA to exploit data parallelism. Numerical results of reflection error and efficiency gain will be presented.

3.6.1 ADI-FDTD Method with CFS-CPML

We first write the conventional ADI-FDTD method with CFS-CPML [47] in compact matrix form as

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}(\mathbf{A}'_E + \frac{\mathbf{L}}{2})\right)\mathbf{u}^{n+\frac{1}{2}} = \left(\mathbf{I}_6 + \frac{\Delta t}{2}(\mathbf{B}'_E + \frac{\mathbf{L}}{2})\right)\mathbf{u}^n + \frac{\Delta t}{2}\mathbf{W}\hat{\Psi}^n \quad (3.6.1a)$$

$$\hat{\Psi}^{n+\frac{1}{2}} = \mathbf{C}\hat{\Psi}^n + \mathbf{D}\mathbf{u}^{n+\frac{1}{2}} \quad (3.6.1b)$$

$$\left(\mathbf{I}_6 - \frac{\Delta t}{2}(\mathbf{B}'_E + \frac{\mathbf{L}}{2})\right)\mathbf{u}^{n+1} = \left(\mathbf{I}_6 + \frac{\Delta t}{2}(\mathbf{A}'_E + \frac{\mathbf{L}}{2})\right)\mathbf{u}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\mathbf{W}\hat{\Psi}^{n+\frac{1}{2}} \quad (3.6.1c)$$

$$\hat{\Psi}^{n+1} = \mathbf{C}\hat{\Psi}^{n+\frac{1}{2}} + \mathbf{D}\mathbf{u}^{n+1} \quad (3.6.1d)$$

where

$$\mathbf{A}'_E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{\partial_y}{\epsilon\hat{\kappa}_y} \\ 0 & 0 & 0 & \frac{\partial_z}{\epsilon\hat{\kappa}_z} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial_x}{\epsilon\hat{\kappa}_x} & 0 \\ 0 & \frac{\partial_z}{\mu\hat{\kappa}_z} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial_x}{\mu\hat{\kappa}_x} & 0 & 0 & 0 \\ \frac{\partial_y}{\mu\hat{\kappa}_y} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}'_E = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{-\partial_z}{\epsilon\hat{\kappa}_z} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{-\partial_x}{\epsilon\hat{\kappa}_x} \\ 0 & 0 & 0 & \frac{-\partial_y}{\epsilon\hat{\kappa}_y} & 0 & 0 \\ 0 & 0 & \frac{-\partial_x}{\mu\hat{\kappa}_x} & 0 & 0 & 0 \\ \frac{-\partial_z}{\mu\hat{\kappa}_z} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-\partial_y}{\mu\hat{\kappa}_y} & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{L} = \begin{bmatrix} -\frac{\sigma}{\epsilon} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\sigma}{\epsilon} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\sigma}{\epsilon} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sigma^m}{\mu} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{\sigma^m}{\mu} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{\sigma^m}{\mu} \end{bmatrix},$$

$$\mathbf{W} = \begin{bmatrix} \frac{1}{\epsilon}\Theta & \mathbf{O}_{3 \times 6} \\ \mathbf{O}_{3 \times 6} & \frac{1}{\mu}\Theta \end{bmatrix}, \Theta = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_e & \mathbf{O}_{6 \times 6} \\ \mathbf{O}_{6 \times 6} & \mathbf{C}_h \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \mathbf{O}_{6 \times 3} & \mathbf{D}_e \\ \mathbf{D}_h & \mathbf{O}_{6 \times 3} \end{bmatrix}, \hat{\Psi} = [\hat{\Psi}_E \quad \hat{\Psi}_H]^T,$$

$$\begin{aligned}
\mathbf{C}_e &= \begin{bmatrix} \check{c}_y & 0 & 0 & 0 & 0 & 0 \\ 0 & \check{c}_z & 0 & 0 & 0 & 0 \\ 0 & 0 & \check{c}_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \check{c}_x & 0 & 0 \\ 0 & 0 & 0 & 0 & \check{c}_x & 0 \\ 0 & 0 & 0 & 0 & 0 & \check{c}_y \end{bmatrix}, \mathbf{D}_e = \begin{bmatrix} 0 & 0 & \check{b}_y \partial_y \\ 0 & \check{b}_z \partial_z & 0 \\ \check{b}_z \partial_z & 0 & 0 \\ 0 & 0 & \check{b}_x \partial_x \\ 0 & \check{b}_x \partial_x & 0 \\ \check{b}_y \partial_y & 0 & 0 \end{bmatrix}, \\
\mathbf{C}_h &= \begin{bmatrix} \check{c}_z & 0 & 0 & 0 & 0 & 0 \\ 0 & \check{c}_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \check{c}_x & 0 & 0 & 0 \\ 0 & 0 & 0 & \check{c}_z & 0 & 0 \\ 0 & 0 & 0 & 0 & \check{c}_y & 0 \\ 0 & 0 & 0 & 0 & 0 & \check{c}_x \end{bmatrix}, \mathbf{D}_h = \begin{bmatrix} 0 & \check{b}_z \partial_z & 0 \\ 0 & 0 & \check{b}_y \partial_y \\ 0 & 0 & \check{b}_x \partial_x \\ \check{b}_z \partial_z & 0 & 0 \\ \check{b}_y \partial_y & 0 & 0 \\ 0 & \check{b}_x \partial_x & 0 \end{bmatrix}, \\
\widehat{\Psi}_E &= \begin{bmatrix} \psi_{e_{xy}} & \psi_{e_{xz}} & \psi_{e_{yz}} & \psi_{e_{yx}} & \psi_{e_{zx}} & \psi_{e_{zy}} \end{bmatrix}, \\
\widehat{\Psi}_H &= \begin{bmatrix} \psi_{h_{xz}} & \psi_{h_{xy}} & \psi_{h_{yx}} & \psi_{h_{yz}} & \psi_{h_{zy}} & \psi_{h_{zx}} \end{bmatrix}.
\end{aligned}$$

$\mathbf{O}_{p \times q}$ is the null matrix with $p \times q$ dimension, σ and σ^m are electric and magnetic conductivities, respectively. Note that σ is different from σ_ξ in (2.1.35), with the latter referring to the conductivity profile of the PML. $\check{c}_\xi$ and $\check{b}_\xi$ are update coefficients for $\widehat{\Psi}$ in the PML regions defined as

$$\check{c}_\xi = e^{-\left(\frac{\sigma_\xi}{\widehat{\kappa}_\xi} + \widehat{\alpha}_\xi\right) \frac{\Delta t}{2}}, \quad (3.6.2)$$

$$\check{b}_\xi = \frac{\sigma_\xi}{\sigma_\xi \widehat{\kappa}_\xi + \widehat{\kappa}_\xi^2 \widehat{\alpha}_\xi} (\check{c}_\xi - 1). \quad (3.6.3)$$

Assuming $\sigma^m = 0$, by expanding (3.6.1) with some manipulations and arrangements, the update equations for the conventional ADI-FDTD method with CFS-CPML are as follows (other field equations can be written down by permuting the indices):

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned}
& -\frac{a_{1,y}a_{2,y}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{y_{j-\frac{1}{2}}}\beta}E_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} - \frac{a_{1,y}a_{2,y}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{y_{j+\frac{1}{2}}}\beta}E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} + \gamma_{y_j}E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \\
& = \frac{\chi}{\beta}E_x|_{i+\frac{1}{2},j,k}^n - \frac{a_{1,z}}{\widehat{\kappa}_{z_k}\beta}\left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^n\right) \\
& - \frac{a_{1,y}a_{2,x}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{x_{i+\frac{1}{2}}}\beta}\left(E_y|_{i+1,j+\frac{1}{2},k}^n - E_y|_{i,j+\frac{1}{2},k}^n - E_y|_{i+1,j-\frac{1}{2},k}^n + E_y|_{i,j-\frac{1}{2},k}^n\right) \\
& + \frac{a_{1,y}}{\widehat{\kappa}_{y_j}\beta}\left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) + \frac{a_1}{\beta}\left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n\right) \\
& + \frac{a_{1,y}a_2}{\widehat{\kappa}_{y_j}\beta}\left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zy}}|_{i+\frac{1}{2},j-\frac{1}{2},k}^n + \psi_{h_{zx}}|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) \quad (3.6.4a)
\end{aligned}$$

$$\begin{aligned}
H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} & = H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}}\left(E_y|_{i+1,j+\frac{1}{2},k}^n - E_y|_{i,j+\frac{1}{2},k}^n\right) \\
& + \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}}\left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}\right) + a_2\left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n\right) \quad (3.6.4b)
\end{aligned}$$

with the update equations of discrete variables $\psi_e^{n+\frac{1}{2}}$ and $\psi_h^{n+\frac{1}{2}}$ as

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{y_j}\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{y_j}}{\Delta y}\left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}\right) \quad (3.6.5a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{z_k}\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{z_k}}{\Delta z}\left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}\right) \quad (3.6.5b)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{y_{j+\frac{1}{2}}}\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{\Delta y}\left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}\right) \quad (3.6.5c)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{x_{i+\frac{1}{2}}}\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{\Delta x}\left(E_y|_{i+1,j+\frac{1}{2},k}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}}\right) \quad (3.6.5d)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned}
& -\frac{a_{1,z}a_{2,z}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{z_{k-\frac{1}{2}}}\beta}E_x|_{i+\frac{1}{2},j,k-1}^{n+1} - \frac{a_{1,z}a_{2,z}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{z_{k+\frac{1}{2}}}\beta}E_x|_{i+\frac{1}{2},j,k+1}^{n+1} + \gamma_{z_k}E_x|_{i+\frac{1}{2},j,k}^{n+1} \\
& = \frac{\chi}{\beta}E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{a_{1,y}}{\widehat{\kappa}_{y_j}\beta}\left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}\right) \\
& - \frac{a_{1,z}a_{2,x}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{x_{i+\frac{1}{2}}}\beta}\left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i+1,j,k-\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,j,k-\frac{1}{2}}^{n+\frac{1}{2}}\right) \\
& - \frac{a_{1,z}}{\widehat{\kappa}_{z_k}\beta}\left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}\right) + \frac{a_1}{\beta}\left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}\right) \\
& - \frac{a_{1,z}a_2}{\widehat{\kappa}_{z_k}\beta}\left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{yx}}|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} + \psi_{h_{yz}}|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}\right) \quad (3.6.6a)
\end{aligned}$$

$$\begin{aligned}
H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} &= H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \\
&+ \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (3.6.6b)
\end{aligned}$$

with the update equations of discrete variables ψ_e^{n+1} and ψ_h^{n+1} as

$$\psi_{exy}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{y_j} \psi_{exy}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_j}}{\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+1} \right) \quad (3.6.7a)$$

$$\psi_{exz}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{z_k} \psi_{exz}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{z_k}}{\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+1} \right) \quad (3.6.7b)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (3.6.7c)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (3.6.7d)$$

where

$$\begin{aligned}
a_{1,\xi} &= \frac{a_1}{\Delta \xi}, \quad a_{2,\xi} = \frac{a_2}{\Delta \xi}, \quad \gamma_{x_i} = 1 + \frac{a_{1,x} a_{2,x}}{\widehat{\kappa}_{x_k} \widehat{\kappa}_{x_{i+\frac{1}{2}}}} \beta + \frac{a_{1,x} a_{2,x}}{\widehat{\kappa}_{x_i} \widehat{\kappa}_{x_{i-\frac{1}{2}}}} \beta, \\
\beta &= \left(1 + a_1 \frac{\sigma}{2} \right), \quad \gamma_{y_j} = 1 + \frac{a_{1,y} a_{2,y}}{\widehat{\kappa}_{y_j} \widehat{\kappa}_{y_{j+\frac{1}{2}}}} \beta + \frac{a_{1,y} a_{2,y}}{\widehat{\kappa}_{y_j} \widehat{\kappa}_{y_{j-\frac{1}{2}}}} \beta, \\
\chi &= \left(1 - a_1 \frac{\sigma}{2} \right), \quad \gamma_{z_k} = 1 + \frac{a_{1,z} a_{2,z}}{\widehat{\kappa}_{z_k} \widehat{\kappa}_{z_{k+\frac{1}{2}}}} \beta + \frac{a_{1,z} a_{2,z}}{\widehat{\kappa}_{z_k} \widehat{\kappa}_{z_{k-\frac{1}{2}}}} \beta.
\end{aligned}$$

Note that for simplicity, we omit the subscript indices for media parameters ϵ , μ and σ . The full update equations of the conventional 3-D ADI-FDTD method with CFS-CPML are in Appendix F.

3.6.2 FADI-FDTD Method with CFS-CPML

The conventional ADI-FDTD with CFS-CPML still involves matrix operators $\mathbf{A}'_E$ and $\mathbf{B}'_E$ on the RHS (c.f. (3.6.1a) and (3.6.1c)). To remove $\mathbf{A}'_E$ and $\mathbf{B}'_E$ from the RHS, we formulate the FADI-FDTD scheme as

$$\mathbf{v}_E^n = \tilde{\mathbf{u}}^n - \mathbf{v}_E^{n-\frac{1}{2}} + \frac{\Delta t}{2} \mathbf{W} \widehat{\Psi}^n \quad (3.6.8a)$$

$$\left(\frac{1}{2}\mathbf{I}_6 - \frac{\Delta t}{4}(\mathbf{A}'_E + \frac{\mathbf{L}}{2})\right)\tilde{\mathbf{u}}^{n+\frac{1}{2}} = \mathbf{v}_E^n \quad (3.6.8b)$$

$$\hat{\Psi}^{n+\frac{1}{2}} = \mathbf{C}\hat{\Psi}^n + \frac{\mathbf{D}}{2}\tilde{\mathbf{u}}^{n+\frac{1}{2}} \quad (3.6.8c)$$

$$\mathbf{v}_E^{n+\frac{1}{2}} = \tilde{\mathbf{u}}^{n+\frac{1}{2}} - \mathbf{v}_E^n + \frac{\Delta t}{2}\mathbf{W}\hat{\Psi}^{n+\frac{1}{2}} \quad (3.6.8d)$$

$$\left(\frac{1}{2}\mathbf{I}_6 - \frac{\Delta t}{4}(\mathbf{B}'_E + \frac{\mathbf{L}}{2})\right)\tilde{\mathbf{u}}^{n+1} = \mathbf{v}_E^{n+\frac{1}{2}} \quad (3.6.8e)$$

$$\hat{\Psi}^{n+1} = \mathbf{C}\hat{\Psi}^{n+\frac{1}{2}} + \frac{\mathbf{D}}{2}\tilde{\mathbf{u}}^{n+1} \quad (3.6.8f)$$

with initialization $\mathbf{v}_E^{-\frac{1}{2}} = \left(\mathbf{I}_6 - \frac{\Delta t}{2}(\mathbf{B}'_E + \frac{\mathbf{L}}{2})\right)\mathbf{u}^0$ in the main grid. Now it can be seen that the algorithm has its RHS free of matrix operators $\mathbf{A}'_E$ and $\mathbf{B}'_E$. This results in the reduction of the number of update coefficients and field variables.

Assuming $\sigma^m = 0$, by expanding (3.6.8), upon some manipulations and arrangements, the update equations for the FADI-FDTD method with CFS-CPML are as follows (other field equations can be written down by permuting the indices):

For first procedure from n to $n + \frac{1}{2}$:

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j,k}^n = \tilde{E}_x|_{i+\frac{1}{2},j,k}^n - e_x|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} + a_1\left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n\right) \quad (3.6.9a)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n = \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n-\frac{1}{2}} + a_2\left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n\right) \quad (3.6.9b)$$

b) Implicit update for $\tilde{E}$

$$\begin{aligned} & -\frac{a_{1,y}a_{2,y}}{2\hat{\kappa}_{y_j}\hat{\kappa}_{y_{j-\frac{1}{2}}}\beta}\tilde{E}_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} - \frac{a_{1,y}a_{2,y}}{2\hat{\kappa}_{y_j}\hat{\kappa}_{y_{j+\frac{1}{2}}}\beta}\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} + \frac{\gamma_{y_j}}{2}\tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \\ & = \frac{1}{\beta}e_x|_{i+\frac{1}{2},j,k}^n + \frac{a_{1,y}}{\hat{\kappa}_{y_j}\beta}\left(h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n\right) \end{aligned} \quad (3.6.10a)$$

c) Explicit update for $\tilde{H}$

$$\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = 2h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{a_{2,y}}{\hat{\kappa}_{y_{j+\frac{1}{2}}}}\left(\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}}\right) \quad (3.6.11a)$$

d) Explicit update for ψ_e and ψ_h

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{y_j}}{2\Delta y} \left(\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \tilde{H}_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (3.6.12a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{z_k}}{2\Delta z} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (3.6.12b)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{2\Delta y} \left(\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (3.6.12c)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{2\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (3.6.12d)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - e_x|_{i+\frac{1}{2},j,k}^n + a_1 \left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (3.6.13a)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + a_2 \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (3.6.13b)$$

b) Implicit update for $\tilde{E}$

$$\begin{aligned} & -\frac{a_{1,z}a_{2,z}}{2\hat{\kappa}_{z_k}\hat{\kappa}_{z_{k-\frac{1}{2}}}\beta} \tilde{E}_x|_{i+\frac{1}{2},j,k-1}^{n+1} - \frac{a_{1,z}a_{2,z}}{2\hat{\kappa}_{z_k}\hat{\kappa}_{z_{k+\frac{1}{2}}}\beta} \tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} + \frac{\gamma_{z_k}}{2} \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \\ & = \frac{1}{\beta} e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{a_{1,z}}{\hat{\kappa}_{z_k}\beta} \left(h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - h_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (3.6.14a)$$

c) Explicit update for $\tilde{H}$

$$\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = 2h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{a_{2,x}}{\hat{\kappa}_{x_{i+\frac{1}{2}}}} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (3.6.15a)$$

d) Explicit update for ψ_e and ψ_h

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_j}}{2\Delta y} \left(\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - \tilde{H}_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+1} \right) \quad (3.6.16a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{z_k}}{2\Delta z} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - \tilde{H}_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+1} \right) \quad (3.6.16b)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{2\Delta y} \left(\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+1} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (3.6.16c)$$

Table 3.8: Flops count for the CFS-PML using conventional and efficient fundamental FADI-FDTD methods

CFS-PML		ADI-FDTD (3.6.1)	FADI-FDTD (3.6.8)
Implicit	A/S	84	12
	M/D	36	12
Explicit	A/S	36	48
	M/D	18	24
ψ_e & ψ_h	A/S	48	48
	M/D	48	48
Total	A/S	168	108
	M/D	102	84
	A/S + M/D	270	192

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{2\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (3.6.16d)$$

For non-zero initial $\tilde{E}^0$ and $\tilde{H}^0$, we apply the initialization as follows:

$$e_x|_{i+\frac{1}{2},j,k}^{-\frac{1}{2}} = \frac{\beta}{2} \tilde{E}_x|_{i+\frac{1}{2},j,k}^0 + \frac{a_{1,z}}{2\hat{\kappa}_{z_k}} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^0 - \tilde{H}_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^0 \right) \quad (3.6.17a)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{-\frac{1}{2}} = \frac{1}{2} \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^0 + \frac{a_{2,x}}{2\hat{\kappa}_{x_{i+\frac{1}{2}}}} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^0 - \tilde{E}_y|_{i,j+\frac{1}{2},k}^0 \right) \quad (3.6.17b)$$

The full update equations of 3-D FADI-FDTD method with CFS-CPML are in Appendix G. In addition, the flops count for the CFS-CPML using conventional and efficient fundamental FADI-FDTD methods is shown in Table 3.8. The comparison of these update equations (inclusive of CFS-CPML) in the conventional and efficient fundamental schemes is based on their respective RHS. From Table 3.8, there is a significant decrease in the total flops count using the FADI-FDTD over conventional ADI-FDTD with CFS-CPML. We can see that in the FADI-FDTD, ψ is only required in the auxiliary field update equations (c.f. (3.6.9) and (3.6.13)) and

Figure 3.6: CUDA Threads Organization for E_x .

NOT in the implicit update equations of electric fields (c.f. (3.6.10) and (3.6.14)). In addition, ψ_e and ψ_h are well separated in the FADI-FDTD update equations (i.e. ψ_e is only required in the electric field update equation while ψ_h is only required in the magnetic field update equation). On the other hand, for the conventional ADI-FDTD, both ψ_e and ψ_h are required simultaneously at the implicit electric field update equations as evident from (3.6.4a) and (3.6.6a). Furthermore, the number of overall RHS terms in the conventional ADI-FDTD update equations is higher compared to that of the FADI-FDTD, which results in more arithmetic and memory indexing operations. For all these advantages, our FADI-FDTD method is very attractive for its better conciseness, efficiency and programming ease. To further increase the efficiency, we incorporate the FADI-FDTD method into the GPU using CUDA to be shown in the next section.

3.6.3 GPU-accelerated FADI-FDTD method with CFS-CPML

We make use of the 2-D grid and one-dimensional (1-D) threads for the field vector E_x field along x -, y -, and z -directions. In the first procedure of the FADI-FDTD

Figure 3.7: Flowchart of the FADI-FDTD method with CFS-CPML as implemented on GPU.

update equations, $E_x|_{i+\frac{1}{2},j+1,k}$ requires the updated value of $E_x|_{i+\frac{1}{2},j,k}$ along the y -direction. This makes E_x along x - and z -directions possible for parallelism. Conversely, in the second procedure, $E_x|_{i+\frac{1}{2},j,k+1}$ requires the updated value of $E_x|_{i+\frac{1}{2},j,k}$ along the z -direction. This in turn makes E_x along x - and y -directions possible for parallelism. The parallelization of the FADI-FDTD method is realized through

Gaussian elimination using the LU Factorization method [107]. In order to exploit the data parallelism in these directions, we have arranged the field vector for E_x in CUDA according to the arrangement shown in Figure 3.6. For instance, in the first procedure, blocks from the same row are run in parallel (data parallelism in x - and z -directions). Starting from the first row, the process will move to the subsequent row after all blocks from the preceding row are processed. This row by row procedure will continue until the maximum number of grids in the y -direction is reached. On the other hand, in the second procedure, blocks from the same column are run in parallel (data parallelism in x - and y -directions). The process is iterated column by column until the maximum number of grids in the z -direction is reached.

The update procedure of the GPU-accelerated FADI-FDTD method with CFS-CPML using CUDA is depicted as a flowchart in Figure 3.7. Note that je and ke are the number of cells in y - and z -directions, respectively. For the update equations of E fields, we only show those of E_x . The remaining procedures for E_y and E_z fields can be obtained by permuting the indices. As mentioned, the ψ 's of the FADI-FDTD method are now included in the auxiliary update equations. Therefore, the main update equations of E fields are the same as the FADI-FDTD method without CFS-CPML, which feature great simplicity and convenience.

3.6.4 Numerical Results

In this section, the performance of CPU and GPU-accelerated FADI-FDTD methods with CFS-CPML in free space is illustrated through numerical simulations. The computation domain has a dimension of $42 \times 42 \times 42$ grids and spatial steps $\Delta x = \Delta y = \Delta z = 1.0$ mm. Ten cells of PML are applied to all six sides of the lattice. Within the PML, the constitutive parameters are scaled according to (2.1.36) and (2.1.37). A choice for $\sigma_{\xi_{max}}$ that will minimize reflection [39] is expressed in (2.1.38).

Figure 3.8: Comparison of time domain waveforms for the electric field at observation point (CFLN = 4).

In our study, the PML parameters are $\sigma_{\xi_{max}} = \sigma_{opt} = 10.61$ S/m, $\kappa_{\xi_{max}} = 15$, $\alpha_{\xi} = 0.08$ S/m, and $m = 4$.

A small dipole source excitation is located at the center of the computation region. The source is a modulated Gaussian pulse applied to E_z component as

$$J_z = e^{-\left(\frac{t-t_0}{\tau}\right)^2} \sin\left(2\pi f_c(t-t_0)\right), \quad (3.6.18)$$

where $\tau = 160$ ps, $t_0 = 3\tau$, $f_c = 3.175$ GHz.

The reflection error for the PML is studied with the observation point located ten cells away from the source and one cell away from the PML interface. The reflection error for the PML is evaluated by

$$R_E = 20 \log_{10} \left(\frac{|E - E_{ref}|}{\max |E_{ref}|} \right) \quad (3.6.19)$$

where E_{ref} is the reference value calculated in a grid large enough so that any reflection from the boundary is isolated.

For comparison, Figure 3.8 presents the electric field's time domain characteristic at the observation point for both CPU and GPU-accelerated FADI-FDTD methods with CFS-CPML. CFLN is chosen as 4, where $\Delta t_{CFL} = 1.926$ ps. The good

Figure 3.9: Reflection errors of CPU and GPU-accelerated FADI-FDTD methods with CFS-CPML for (a) CFLN = 1, (b) CFLN = 1 (magnified), (c) CFLN = 2 and (d) CFLN = 4.

agreement indicates that the GPU-accelerated FADI-FDTD method is correct although the GPU adopts single precision floating-point format as compared to the double-precision floating-point format of the CPU.

Figure 3.9 illustrates the reflection errors of CPU and GPU-accelerated FADI-FDTD methods with CFS-CPML for various CFLNs. Note that Figure 3.9b shows the magnified view of Figure 3.9a for better reading of the graph. The maximum amplitude of $|E_{ref}|$ used in (3.6.19) for CFLN 1, 2 and 4 is given as 0.0655, 0.0691 and 0.0857 V/m, respectively. For CFLN = 1, both FADI- and Yee-FDTD methods with CFS-CPML exhibit reflection errors of the same order. In addition, we observe that the reflection errors computed using both CPU and GPU implementations agree well with each other, and the maximum reflection error is around -75 dB for CFLN = 4. Such low reflection error shows the effectiveness of the CFS-CPML for absorbing outgoing electromagnetic wave. Both CPU and GPU implementations are equally effective, with greater efficiency for the latter. The slight discrepancies of reflection errors are the result of single-precision floating-point format supported by the low cost GPU.

We further compare the CPU and GPU computation time of the FADI-FDTD method with CFS-CPML, c.f. Table 3.9. The numerical simulation is performed for 250 time steps at CFLN = 4 in various computation domains. Note that 250 time steps at CFLN = 4 are more than sufficient for the entire wave motion to reach the PML layer. The computing platform is Microsoft Visual C++ environment with Intel Dual Core 2.66 GHz processor and a NVIDIA GeForce GTS250 GPU. From Table 3.9, we find that the efficiency gain achieved by the GPU over CPU is approximately 11~15 times. It is thus ascertained that substantial gain is achievable for the GPU by exploiting data parallelism. The numerical results for the Yee-FDTD method with CFS-CPML are also included in Table 3.9, for 1000

Table 3.9: Comparison of CPU and GPU computation time for FADI-FDTD and Yee-FDTD methods with CFS-CPML (For the FADI-FDTD, CFLN = 4, 250 time steps. For the Yee-FDTD, CFLN = 1, 1000 time steps)

Scheme	Domain Size	CPU Time (s)	GPU Time (s)	Speedup
FADI-FDTD	$100 \times 100 \times 100$	679.53	60.74	11.19
with	$150 \times 150 \times 150$	2842.11	201.31	14.12
CFS-CPML	$200 \times 200 \times 200$	6828.12	463.88	14.72
Yee-FDTD	$100 \times 100 \times 100$	1011.70	79.92	12.66
with	$150 \times 150 \times 150$	4810.23	263.27	18.27
CFS-CPML	$200 \times 200 \times 200$	11504.21	621.39	18.51

time steps at CFLN = 1. The efficiency gain achieved by the GPU over CPU of the Yee-FDTD method is slightly higher than that of the FADI-FDTD method. This is because the Yee-FDTD is a fully explicit scheme which allow data parallelism in three directions (x , y and z) while for the FADI-FDTD method, data parallelism is in two directions as discussed in Section 3.6.3. Nevertheless, the overall CPU and GPU computation times incurred by the Yee-FDTD method are still higher than that of the FADI-FDTD method, due to its restricted time step caused by the CFL stability condition.

3.6.5 Application of FADI-FDTD method with CFS-CPML

We now simulate a microstrip low-pass filter [108, 109] using our FADI-FDTD method with CFS-CPML shown in Figure 3.10. The computation domain has a dimension of $113 \times 122 \times 36$ together with ten cells of PML applied to all six sides of the lattice. The spatial steps are $\Delta x = \Delta y = \Delta z = 2.0$ mm. To illustrate the non-uniform grids, the thickness of the microstrip $\Delta z = 2.31$ μm is set in one layer above the substrate with $\epsilon = 2.2$. A line electric current source is applied below the

Figure 3.10: Microstrip low-pass filter [109].

microstrip of port 1 and one cell before the PML layer. The reference planes for ports 1 and 2 are located 20 cells away from the edges of the rectangle patch. A differentiated Gaussian pulse is utilized as excitation:

$$J_z = -2 \frac{t - t_0}{\tau} e^{-\left(\frac{t - t_0}{\tau}\right)^2} \quad (3.6.20)$$

where $\tau = 40$ ps, $t_0 = 3\tau$.

Due to the stability constraint of the Yee-FDTD method, the $\Delta t_{CFL} = 7.7$ ps requires a simulation of 3×10^5 time steps to allow the response on both ports to become nearly 0. Using our unconditionally stable FADI-FDTD method that is not constrained by the CFL condition, we can use a large CFLN value to reduce the number of simulation time steps, leading to lower computation time.

Figures 3.11 and 3.12 show the return loss and insertion loss of the low-pass filter respectively using the FADI-FDTD method with CFS-CPML for various CFLNs. For comparison, the Yee-FDTD method with CFS-CPML has been included in both figures. The results are in good agreement with each other. By exploiting the FADI-FDTD method unconditionally stable feature, we can use a high CFLN value (CFLN = 50, 100, 150, 200). This enables the FADI-FDTD method to achieve a high computational power (CPU efficiency gain of 15, 30, 45, 60 times) over the

Figure 3.11: Return loss of the low-pass filter with various CFLNs.

Figure 3.12: Insertion loss of the low-pass filter with various CFLNs.

Yee-FDTD method.

3.7 Conclusions

This chapter has presented the formulation of the conventional ADI- and LOD-FDTD methods into their respective efficient fundamental forms. Subsequently, we have presented the implementations of PMC and PEC boundary conditions for FADI- and FLOD-FDTD methods for closed region simulation. By introducing image theory, the PMC (or PEC) boundary equations for the implicit updating of electric (or magnetic) fields in FADI- and FLOD-FDTD methods have been derived. Image theory facilitates the implementation of PMC and/or PEC boundary conditions regardless of whether the update equations for electric or magnetic fields are implicit. Comparison between the PMC and PEC boundary equations in the conventional and efficient fundamental ADI- and LOD-FDTD methods has signified a reduction in the flops count, thus justifying the higher efficiency and simplicity for the efficient fundamental schemes.

Next, for open structure simulation, this chapter has presented the implementation of the Mur first order ABC in 3-D FADI- and FLOD-FDTD methods. The Mur ABC has been incorporated into FADI- and FLOD-FDTD methods using consistent implementation and a novel implementation with lower reflection coefficient. Both implementations in FADI- and FLOD-FDTD methods have been compared and validated with its conventional ADI- and LOD-FDTD methods. Comparing the CPU computation time of both conventional and efficient fundamental ADI- and LOD-FDTD methods, it is ascertained that substantial gain in the overall efficiency has been achieved for the latter even with Mur ABC incorporated.

For better absorption (lower reflection coefficient), this chapter has presented the implementation of the split-field PML for the FADI-FDTD method. The split-field PML has been formulated into the ADI generalized splitting formulae cast into matrix-operator-free RHS. The FADI-FDTD method with PML incorporated

is compared to the conventional ADI-FDTD implementation through numerical simulations. It is concluded that their reflection errors are identical and hence confirmed that both implementations are equivalently effective. On the other hand, by comparing their CPU computation time, it is ascertained that substantial gain in the overall efficiency could be achieved for the efficient fundamental scheme.

Lastly, this chapter has presented the GPU-accelerated FADI-FDTD method with CFS-CPML. The compact matrix form of the conventional ADI-FDTD method with CFS-CPML has been formulated into the FADI-FDTD method which leads to a reduction of the number of update coefficients and field variables. Using CUDA, the FADI-FDTD method with CFS-CPML has been further incorporated into the GPU to exploit data parallelism. Numerical results have validated that a much higher efficiency gain of up to 15 times can be achieved. To demonstrate the usefulness of the FADI-FDTD method with CFS-CPML, a practical microstrip low-pass filter has been presented. A high computational power has been attained while preserving a good agreement with the Yee-FDTD method.

Chapter 4

Extension of Stable and Efficient ADI/LOD-FDTD Methods for Heat Transfer

4.1 Introduction

As technology advances in pushing for higher speed and performance, power and packaging densities rise significantly. Since power dissipation is directly translated into heat, the temperature of processors is getting warmer. High temperature causes the transistors to slow down due to the degradation of carrier mobility. The interconnect metal also results in higher resistivity, causing longer interconnect RC delays. All these will translate into degradation in performance for the otherwise high performance chips, while the reliability of the chip will also suffer as increased temperature will decrease the lifespan of the chip [110–113]. It is crucial for chip-level thermal analysis to efficiently analyze the thermal distribution and locate the hot spots. In addition, finite thermal conductivity of the complicated packaging with uniform heat distribution does not guarantee uniform temperature profile. Thus, it is important to know the temperature profile and hot spots, not only for steady state but also for transient state.

Recently, 3-D stacking is drawing a great deal of attention in embedded processor systems [114–118]. It eliminates wires within a microprocessor chip and between disparate chips. In computing systems, the performance, latency and power overhead are attributed primarily by the wires. Therefore, by implementing 3-D stacking, different types of chips can be stacked with a higher bandwidth, lower latency and lower power interface. In addition, wire elimination using 3-D stacking provides new microarchitecture opportunities to trade off performance, power and area.

Various numerical methods [119–135] have been proposed for solving the heat transfer equation. For instance, the modified hybrid Laplace transform finite element method in [119] and the Galerkin gradient least-squares (GGLS) method in [122] have been developed for solving heat conduction problems. A generalized integral transform has been developed in [129] to estimate the temperature distribution of a chip. In [131], the finite volume method for cylindrical heat conduction problems based on local analytical solution has been developed. The mixed-collocation, finite-difference method has been employed in [135] to solve the heat transfer equation for a dual phase lag (DPL) model.

One of the techniques to solve the heat transfer equation is known to be the FDTD method [55]. In this chapter, based on the FDTD technique used for solving Maxwell's equations (which are hyperbolic partial differential equations) in Chapter 3, it can be extended to solve the heat transfer equation (which is a parabolic partial differential equation). The FDTD method has been used due to its simplicity and flexibility in which the algorithm can be either explicit or implicit in nature. However, the explicit finite-difference method is limited by its stability constraint which imposes a maximum limit on the chosen time step depending on the minimum spatial step of the computation domain. To overcome this limitation, an unconditionally stable ADI method based on Douglas-Gunn [60] and Peaceman-

Rachford [61] algorithm has been proposed in [63,64] for the heat transfer equation. Unfortunately, this method is incomplete because it only takes into consideration the Laplacian terms of the heat transfer equation but in fact, the heat transfer equation consists of both Laplacian and gradient terms within general inhomogeneous media. The conventional DG- and PR-ADI methods are no longer unconditionally stable if both Laplacian and gradient terms are present. The instability of the conventional DG- and PR-ADI methods arises due to the presence of the gradient terms in the heat transfer equation when considering inhomogeneous media. Moreover, the ADI method involves a tridiagonal system of equations, with its RHS having a considerable amount of arithmetic operations. This leads to a high memory indexing, increased programming complexity, as well as longer CPU computation time.

To mitigate the high complexity of implicit methods in electromagnetics, efficient algorithms based on the efficient fundamental ADI and LOD methods have been developed in Section 3.2. Nonetheless, the conventional DG-ADI method in heat transfer and the ADI method in electromagnetics differ in terms of their update procedures and operators. The former is not directly reducible to its fundamental form and it remains unclear on how to extend the FADI concept from electromagnetics to heat transfer. Furthermore, as mentioned earlier, the conventional DG-ADI method is not unconditionally stable and prior efforts are needed to ensure its stability before any improvement on the efficiency can be made.

In this chapter, we present a stable 2-D FADI method for solving the heat transfer equation. The formulation will consider both Laplacian and gradient terms of the heat transfer equation. The potential instability of the conventional DG-ADI method caused by the gradient terms within inhomogeneous media is first alleviated. The proposed stabilized DG-ADI method is then cast into the (stabilized) PR-ADI method in compact form, and further formulated into the FADI method

with operator-free RHS, resulting in simpler and more concise update equations. The relationship among temperatures resulted from these three methods, namely, DG-ADI, PR-ADI and FADI methods will be illustrated and discussed in the following sections. Stability analysis by means of analyzing the eigenvalues of the reduced amplification matrix shall be performed to verify the stability of the FADI method, while the potential instability of the conventional DG-ADI method for inhomogeneous media will be demonstrated. Furthermore, the efficiency gain in terms of the CPU computation time for the FADI method over DG-ADI and PR-ADI methods will be presented.

Next, we present the GPU-accelerated FADI method for efficient thermal simulation of ICs with microchannel cooling [136–138]. The convection heat flux due to fluid motion is first included in the heat transfer equation. Subsequently, the FADI method is introduced for thermal simulation with its RHS free of operators, leading to computationally efficient update equations. To further accelerate the FADI method, the GPU will be utilized through CUDA implementation. It will be shown that high efficiency gain can be achieved through large time step size and data parallelism. Numerical results on the cooling effect of the microchannels will also be demonstrated.

The conventional 3-D DG-ADI method is stable in homogeneous media, but it still comes with a high complexity and increased memory variables in its implementation. To overcome these complications, we present a 3-D efficient ADI method for solving the heat transfer equation. The DG-ADI method is formulated into the efficient ADI method with single operator and heat generation input on the RHS of the first procedure, reducing the number of arithmetic operations to the minimal. The efficient ADI method is compared and validated with the DG-ADI method through numerical simulations. Comparison in terms of memory allocation and efficiency

gain for both efficient ADI and DG-ADI methods will be shown and discussed.

Unfortunately, extending the proposed stabilized 2-D DG-ADI method into 3-D DG-ADI method, the potential instability caused by the gradient terms within inhomogeneous media cannot be resolved. To alleviate this problem, we shall present two stable 3-D FLOD methods for thermal simulation within general inhomogeneous media. The first stable LOD method for solving the heat transfer equation takes into account both Laplacian and gradient terms. On the other hand, the second stable LOD method involves the discretization of the heat transfer equation with gradient terms being absorbed into the finite-difference operator directly without being expanded out. The LOD method is then cast into the compact form and formulated into the FLOD method with operator-free RHS, leading to computationally efficient update equations. Stability analysis by means of analyzing the eigenvalues of the amplification matrix will be performed to substantiate the stability of the FLOD method. Additionally, the potential instability of the conventional DG-ADI method for inhomogeneous media will be described. This is followed by the illustration of the efficiency gain of the FLOD method over LOD method as well as the relative maximum error of the FLOD method. To demonstrate the usefulness of the FLOD method, the heat distribution of the closely resembled Alpha 21364 processor chip will be presented and analyzed.

4.2 2-D ADI Method

The temperature of a system is governed by the partial differential equation of the heat transfer equation [52, 53] in (2.2.1). In Chapter 2.2, the simulation of the temperature distribution considered homogeneous material, therefore the term $\nabla \cdot [\kappa(\vec{r}) \nabla T(\vec{r}, t)]$ in (2.2.1) can be replaced by $\kappa(\vec{r}) \nabla^2 T(\vec{r}, t)$. By including the inhomogeneous materials, the term $\nabla \cdot [\kappa(\vec{r}) \nabla T(\vec{r}, t)]$ in (2.2.1) is now expanded

into

$$\rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial t} = \kappa(\vec{r})\nabla^2 T(\vec{r}, t) + \nabla\kappa(\vec{r}) \cdot \nabla T(\vec{r}, t) + g(\vec{r}, t). \quad (4.2.1)$$

One can find both Laplacian $\kappa(\vec{r})\nabla^2 T(\vec{r}, t)$ and gradient $\nabla\kappa(\vec{r}) \cdot \nabla T(\vec{r}, t)$ terms within general inhomogeneous media.

The explicit method for inhomogeneous materials now has the stability constraints [139]

$$\gamma = \vartheta_{i,j}\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \leq \frac{1}{2}. \quad (4.2.2)$$

$$\Delta t \leq \frac{4\vartheta_{i,j}}{(|\hat{u}_{i,j}| + |\hat{v}_{i,j}|)^2} \quad (4.2.3)$$

where

$$\vartheta_{i,j} = \frac{\kappa_{i,j}}{\rho_{i,j}C_{p_{i,j}}}, \quad \hat{u}_{i,j} = \frac{\partial\kappa_{i,j}}{\partial x} \quad \text{and} \quad \hat{v}_{i,j} = \frac{\partial\kappa_{i,j}}{\partial y}.$$

The constraints in (4.2.2) and (4.2.3) arise from the Laplacian and gradient terms respectively. This restricts the time step Δt for certain preset inhomogeneities and spatial steps of Δx and Δy . Therefore, the application of the explicit method becomes increasingly difficult especially when one or both of the constraints become too restrictive.

By using the conventional DG-ADI method, (4.2.1) can be solved in two procedures with the inclusion of both Laplacian and gradient terms as

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} &= \frac{r_{i,j}}{2} \left[\kappa_{i,j} \frac{\partial^2}{\partial x^2} + \frac{\partial\kappa_{i,j}}{\partial x} \frac{\partial}{\partial x} \right] (T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n) \\ &\quad + r_{i,j} \left[\kappa_{i,j} \frac{\partial^2}{\partial y^2} + \frac{\partial\kappa_{i,j}}{\partial y} \frac{\partial}{\partial y} \right] T_{\text{DG}}|_{i,j}^n + T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \end{aligned} \quad (4.2.4a)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} T_{\text{DG}}|_{i,j}^{n+1} = & \frac{r_{i,j}}{2} \left[\kappa_{i,j} \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j}}{\partial x} \frac{\partial}{\partial x} \right] (T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n) \\ & + \frac{r_{i,j}}{2} \left[\kappa_{i,j} \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j}}{\partial y} \frac{\partial}{\partial y} \right] (T_{\text{DG}}|_{i,j}^{n+1} + T_{\text{DG}}|_{i,j}^n) + T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \end{aligned} \quad (4.2.4b)$$

where

$$\begin{aligned} \frac{\partial^2 T_{\text{DG}}|_{i,j}^n}{\partial x^2} &= \frac{T_{\text{DG}}|_{i+1,j}^n - 2T_{\text{DG}}|_{i,j}^n + T_{\text{DG}}|_{i-1,j}^n}{\Delta x^2}, \quad \frac{\partial T_{\text{DG}}|_{i,j}^n}{\partial x} = \frac{T_{\text{DG}}|_{i+1,j}^n - T_{\text{DG}}|_{i-1,j}^n}{2\Delta x}, \\ \frac{\partial^2 T_{\text{DG}}|_{i,j}^n}{\partial y^2} &= \frac{T_{\text{DG}}|_{i,j+1}^n - 2T_{\text{DG}}|_{i,j}^n + T_{\text{DG}}|_{i,j-1}^n}{\Delta y^2}, \quad \frac{\partial T_{\text{DG}}|_{i,j}^n}{\partial y} = \frac{T_{\text{DG}}|_{i,j+1}^n - T_{\text{DG}}|_{i,j-1}^n}{2\Delta y}, \\ \frac{\partial \kappa_{i,j}}{\partial x} &= \frac{\kappa_{i+1,j} - \kappa_{i-1,j}}{2\Delta x}, \quad \frac{\partial \kappa_{i,j}}{\partial y} = \frac{\kappa_{i,j+1} - \kappa_{i,j-1}}{2\Delta y}, \quad r_{i,j} = \frac{\Delta t}{\rho_{i,j} C_{p_{i,j}}}, \quad G|_{i,j}^n = \frac{\Delta t}{\rho_{i,j} C_{p_{i,j}}} g|_{i,j}^n. \end{aligned}$$

It is worth noting that this method can only remove the stability constraint in (4.2.2) due to the Laplacian terms. To the best of the author's knowledge, many researchers omit the gradient terms in the applications of conventional DG-ADI method in (4.2.4), thus exploiting its (apparent) unconditional stability. However, if both Laplacian and gradient terms are included (as required) in the heat transfer equation (2.2.1), the stability constraint in (4.2.3) due to the gradient terms may still cause potential instability (to be demonstrated in Section 4.2.3). Similar potential instability may be present in the conventional PR-ADI method.

4.2.1 Proposed Stabilized DG-ADI and PR-ADI Methods

To eliminate both stability constraints of (4.2.2) and (4.2.3) completely, we introduce a spatial averaging operator $\square_{\bar{\xi}}$ to κ in (4.2.4). The proposed DG-ADI method calls for the following update procedures:

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} = & \frac{r_{i,j}}{2} \left[(\square_x \kappa_{i,j}) \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j}}{\partial x} \frac{\partial}{\partial x} \right] (T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n) \\ & + r_{i,j} \left[(\square_y \kappa_{i,j}) \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j}}{\partial y} \frac{\partial}{\partial y} \right] T_{\text{DG}}|_{i,j}^n + T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \end{aligned} \quad (4.2.5a)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} T_{\text{DG}}|_{i,j}^{n+1} &= \frac{r_{i,j}}{2} \left[(\square_x \kappa_{i,j}) \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j}}{\partial x} \frac{\partial}{\partial x} \right] (T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n) \\ &\quad + \frac{r_{i,j}}{2} \left[(\square_y \kappa_{i,j}) \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j}}{\partial y} \frac{\partial}{\partial y} \right] (T_{\text{DG}}|_{i,j}^{n+1} + T_{\text{DG}}|_{i,j}^n) + T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \end{aligned} \quad (4.2.5b)$$

where

$$\square_x \kappa_{i,j} = \frac{\kappa_{i+1,j} + \kappa_{i-1,j}}{2}, \quad \square_y \kappa_{i,j} = \frac{\kappa_{i,j+1} + \kappa_{i,j-1}}{2}.$$

We rearrange (4.2.5) into compact form as

$$\left(1 - \frac{1}{2}A_T\right) T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} = \left(1 + \frac{1}{2}A_T + B_T\right) T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \quad (4.2.6a)$$

$$\left(1 - \frac{1}{2}B_T\right) T_{\text{DG}}|_{i,j}^{n+1} = \left(1 + \frac{1}{2}A_T + \frac{1}{2}B_T\right) T_{\text{DG}}|_{i,j}^n + \frac{1}{2}A_T T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + G|_{i,j}^n \quad (4.2.6b)$$

where

$$A_T = r_{i,j} \left[(\square_x \kappa_{i,j}) \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j}}{\partial x} \frac{\partial}{\partial x} \right], \quad B_T = r_{i,j} \left[(\square_y \kappa_{i,j}) \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j}}{\partial y} \frac{\partial}{\partial y} \right]. \quad (4.2.7)$$

By applying central approximation and upon some manipulations, we obtain

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} &-\frac{1}{2}\alpha_{x,i,j} T_{\text{DG}}|_{i-1,j}^{n+\frac{1}{2}} + (1 + a_{x,i,j}) T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} - \frac{1}{2}\beta_{x,i,j} T_{\text{DG}}|_{i+1,j}^{n+\frac{1}{2}} \\ &= \frac{1}{2}\alpha_{x,i,j} T_{\text{DG}}|_{i-1,j}^n + \frac{1}{2}\beta_{x,i,j} T_{\text{DG}}|_{i+1,j}^n + \alpha_{y,i,j} T_{\text{DG}}|_{i,j-1}^n + \beta_{y,i,j} T_{\text{DG}}|_{i,j+1}^n \\ &\quad + (1 - a_{x,i,j} - 2a_{y,i,j}) T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \end{aligned} \quad (4.2.8a)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} &-\frac{1}{2}\alpha_{y,i,j} T_{\text{DG}}|_{i,j-1}^{n+1} + (1 + a_{y,i,j}) T_{\text{DG}}|_{i,j}^{n+1} - \frac{1}{2}\beta_{y,i,j} T_{\text{DG}}|_{i,j+1}^{n+1} \\ &= \frac{1}{2}\alpha_{x,i,j} \left(T_{\text{DG}}|_{i-1,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i-1,j}^n \right) + \frac{1}{2}\beta_{x,i,j} \left(T_{\text{DG}}|_{i+1,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i+1,j}^n \right) \\ &\quad + \frac{1}{2}\alpha_{y,i,j} T_{\text{DG}}|_{i,j-1}^n + \frac{1}{2}\beta_{x,i,j} T_{\text{DG}}|_{i,j+1}^n + (1 - a_{x,i,j} - a_{y,i,j}) T_{\text{DG}}|_{i,j}^n - a_{x,i,j} T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + G|_{i,j}^n \end{aligned} \quad (4.2.8b)$$

where

$$\alpha_{\tilde{\xi},j} = a_{\tilde{\xi},j} - b_{\tilde{\xi},j}, \quad \beta_{\tilde{\xi},j} = a_{\tilde{\xi},j} + b_{\tilde{\xi},j}, \quad a_{\tilde{\xi},j} = \frac{r_{i,j}}{\Delta \tilde{\xi}^2} \square_{\tilde{\xi}} \kappa_{i,j}, \quad b_{\tilde{\xi},j} = \frac{r_{i,j}}{2\Delta \tilde{\xi}} \frac{\partial \kappa_{i,j}}{\partial \tilde{\xi}}.$$

The proposed DG-ADI method is now stabilized for the heat transfer equation including both Laplacian and gradient terms. The stability analysis of our proposed DG-ADI method will be presented and discussed in Section 4.2.3.

Upon attaining a stabilized DG-ADI method, we find that the RHS of (4.2.8) has a substantial amount of arithmetic operations and is computationally expensive. To reduce the amount of arithmetic operations and its complexity, we modify (4.2.6) into

$$\left(1 - \frac{1}{2}A_T\right)T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} = 2\left(1 + \frac{1}{2}B_T\right)T_{\text{DG}}|_{i,j}^n - \left(1 - \frac{1}{2}A_T\right)T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n \quad (4.2.9a)$$

$$\begin{aligned} \left(1 - \frac{1}{2}B_T\right)T_{\text{DG}}|_{i,j}^{n+1} &= \frac{1}{2}\left(1 + \frac{1}{2}A_T + B_T\right)T_{\text{DG}}|_{i,j}^n + \frac{1}{2}A_T T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} \\ &\quad + \frac{1}{2}\left(1 + \frac{1}{2}A_T\right)T_{\text{DG}}|_{i,j}^n + G|_{i,j}^n. \end{aligned} \quad (4.2.9b)$$

By substituting (4.2.6a) into (4.2.9b) and upon some manipulations, we get

$$\frac{1}{2}\left(1 - \frac{1}{2}A_T\right)(T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n) = \left(1 + \frac{1}{2}B_T\right)T_{\text{DG}}|_{i,j}^n + \frac{1}{2}G|_{i,j}^n \quad (4.2.10a)$$

$$\left(1 - \frac{1}{2}B_T\right)T_{\text{DG}}|_{i,j}^{n+1} = \frac{1}{2}\left(1 + \frac{1}{2}A_T\right)(T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n) + \frac{1}{2}G|_{i,j}^n. \quad (4.2.10b)$$

Through a redefinition of variables in terms of T_{PR} 's, we cast the DG-ADI method into the PR-ADI method [61] in compact form as described below:

$$\left(1 - \frac{1}{2}A_T\right)T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} = \left(1 + \frac{1}{2}B_T\right)T_{\text{PR}}|_{i,j}^n + \frac{1}{2}G|_{i,j}^n \quad (4.2.11a)$$

$$\left(1 - \frac{1}{2}B_T\right)T_{\text{PR}}|_{i,j}^{n+1} = \left(1 + \frac{1}{2}A_T\right)T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} + \frac{1}{2}G|_{i,j}^n. \quad (4.2.11b)$$

In addition, the proposed PR-ADI method is stabilized for the heat transfer equation involving both Laplacian and gradient terms. Here, we note that the temperatures

resulted from both DG-ADI and PR-ADI methods are identical at T^n and T^{n+1} , i.e.

$$T_{\text{PR}}|_{i,j}^n = T_{\text{DG}}|_{i,j}^n \quad (4.2.12a)$$

$$T_{\text{PR}}|_{i,j}^{n+1} = T_{\text{DG}}|_{i,j}^{n+1}. \quad (4.2.12b)$$

However, confusion may arise as the intermediate values $T^{n+\frac{1}{2}}$'s for both DG-ADI and PR-ADI methods are different. In fact, they have the relation

$$T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} = \frac{1}{2}(T_{\text{DG}}|_{i,j}^{n+\frac{1}{2}} + T_{\text{DG}}|_{i,j}^n). \quad (4.2.12c)$$

Finally, applying central approximation and upon some manipulations for both procedures of the PR-ADI method, we have

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} & -\frac{1}{2}\alpha_{x_{i,j}}T_{\text{PR}}|_{i-1,j}^{n+\frac{1}{2}} + (1 + a_{x_{i,j}})T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} - \frac{1}{2}\beta_{x_{i,j}}T_{\text{PR}}|_{i+1,j}^{n+\frac{1}{2}} \\ & = \frac{1}{2}\alpha_{y_{i,j}}T_{\text{PR}}|_{i,j-1}^n + \frac{1}{2}\beta_{y_{i,j}}T_{\text{PR}}|_{i,j+1}^n + (1 - a_{y_{i,j}})T_{\text{PR}}|_{i,j}^n + \frac{1}{2}G|_{i,j}^n \end{aligned} \quad (4.2.13a)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} & -\frac{1}{2}\alpha_{y_{i,j}}T_{\text{PR}}|_{i,j-1}^{n+1} + (1 + a_{y_{i,j}})T_{\text{PR}}|_{i,j}^{n+1} - \frac{1}{2}\beta_{y_{i,j}}T_{\text{PR}}|_{i,j+1}^{n+1} \\ & = \frac{1}{2}\alpha_{x_{i,j}}T_{\text{PR}}|_{i-1,j}^{n+\frac{1}{2}} + \frac{1}{2}\beta_{x_{i,j}}T_{\text{PR}}|_{i+1,j}^{n+\frac{1}{2}} + (1 - a_{x_{i,j}})T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} + \frac{1}{2}G|_{i,j}^n. \end{aligned} \quad (4.2.13b)$$

To prove the stability for the proposed stabilized PR-ADI method, we use the von Neumann method at every grid point of the computation domain. The temperature along the boundary of the domain is set as 0 K. Therefore, (4.2.13) at grid point $i = 2, j = 2$ in Fourier domain can be written as

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} & (1 + a_{x_{2,2}})\tilde{T}_{\text{PR}}^{n+\frac{1}{2}}e^{\mathbf{j}(-2k_x\Delta x - 2k_y\Delta y)} - \frac{1}{2}\beta_{x_{2,2}}\tilde{T}_{\text{PR}}^{n+\frac{1}{2}}e^{\mathbf{j}(-3k_x\Delta x - 2k_y\Delta y)} \\ & = (1 - a_{y_{2,2}})\tilde{T}_{\text{PR}}^ne^{\mathbf{j}(-2k_x\Delta x - 2k_y\Delta y)} + \frac{1}{2}\beta_{y_{2,2}}\tilde{T}_{\text{PR}}^ne^{\mathbf{j}(-2k_x\Delta x - 3k_y\Delta y)} \end{aligned} \quad (4.2.14a)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} & (1 + a_{y_{2,2}}) \tilde{T}_{\text{PR}}^{n+1} e^{\mathbf{j}(-2k_x \Delta x - 2k_y \Delta y)} - \frac{1}{2} \beta_{y_{2,2}} \tilde{T}_{\text{PR}}^{n+1} e^{\mathbf{j}(-2k_x \Delta x - 3k_y \Delta y)} \\ &= (1 - a_{x_{2,2}}) \tilde{T}_{\text{PR}}^{n+\frac{1}{2}} e^{\mathbf{j}(-2k_x \Delta x - 2k_y \Delta y)} + \frac{1}{2} \beta_{x_{2,2}} \tilde{T}_{\text{PR}}^{n+\frac{1}{2}} e^{\mathbf{j}(-3k_x \Delta x - 2k_y \Delta y)}. \end{aligned} \quad (4.2.14b)$$

where $k_{\tilde{\xi}}$ is the spatial frequency along the $\tilde{\xi}$ -direction.

By arranging of the variables and combining the two procedures, we get

$$\begin{aligned} \tilde{T}_{\text{PR}}^{n+1} &= \frac{1 - \left[a_{x_{2,2}} - \frac{1}{2} \beta_{x_{2,2}} (\cos \tilde{\theta}_x - \mathbf{j} \sin \tilde{\theta}_x) \right]}{1 + \left[a_{x_{2,2}} - \frac{1}{2} \beta_{x_{2,2}} (\cos \tilde{\theta}_x - \mathbf{j} \sin \tilde{\theta}_x) \right]} \\ &\quad \cdot \frac{1 - \left[a_{y_{2,2}} - \frac{1}{2} \beta_{y_{2,2}} (\cos \tilde{\theta}_y - \mathbf{j} \sin \tilde{\theta}_y) \right]}{1 + \left[a_{y_{2,2}} - \frac{1}{2} \beta_{y_{2,2}} (\cos \tilde{\theta}_y - \mathbf{j} \sin \tilde{\theta}_y) \right]} \tilde{T}_{\text{PR}}^n \end{aligned} \quad (4.2.15)$$

where $\tilde{\theta}_{\tilde{\xi}} = k_{\tilde{\xi}} \Delta \tilde{\xi}$.

The proposed stabilized PR-ADI method is stable if the amplification matrix is

$$\begin{aligned} & \left| \frac{1 - \left[a_{x_{2,2}} - \frac{1}{2} \beta_{x_{2,2}} \cos \tilde{\theta}_x \right] - \mathbf{j} \frac{1}{2} \beta_{x_{2,2}} \sin \tilde{\theta}_x}{1 + \left[a_{x_{2,2}} - \frac{1}{2} \beta_{x_{2,2}} \cos \tilde{\theta}_x \right] - \mathbf{j} \frac{1}{2} \beta_{x_{2,2}} \sin \tilde{\theta}_x} \right. \\ & \quad \left. \cdot \frac{1 - \left[a_{y_{2,2}} - \frac{1}{2} \beta_{y_{2,2}} \cos \tilde{\theta}_y \right] - \mathbf{j} \frac{1}{2} \beta_{y_{2,2}} \sin \tilde{\theta}_y}{1 + \left[a_{y_{2,2}} - \frac{1}{2} \beta_{y_{2,2}} \cos \tilde{\theta}_y \right] - \mathbf{j} \frac{1}{2} \beta_{y_{2,2}} \sin \tilde{\theta}_y} \right| \leq 1 \end{aligned} \quad (4.2.16)$$

In order for the amplification matrix (4.2.16) to be lesser or equal to 1, $(a_{x_{2,2}} - \frac{1}{2} \beta_{x_{2,2}} \cos \tilde{\theta}_x)$ and $(a_{y_{2,2}} - \frac{1}{2} \beta_{y_{2,2}} \cos \tilde{\theta}_y)$ have to be greater or equal to 0 for all values of $\tilde{\theta}_x$ and $\tilde{\theta}_y$.

Since

$$\begin{aligned} a_{x_{2,2}} - \frac{1}{2} \beta_{x_{2,2}} \cos \tilde{\theta}_x &= \frac{r_{2,2}}{2\Delta x^2} (\kappa_{3,2} + \kappa_{1,2}) - \frac{r_{2,2}}{4\Delta x^2} \left((\kappa_{3,2} + \kappa_{1,2}) + \frac{\kappa_{3,2} - \kappa_{1,2}}{2} \right) \cos \tilde{\theta}_x \\ &= \frac{r_{2,2}}{2\Delta x^2} (\kappa_{3,2} + \kappa_{1,2}) - \frac{r_{2,2}}{4\Delta x^2} \left(\frac{3}{2} \kappa_{3,2} + \frac{1}{2} \kappa_{1,2} \right) \cos \tilde{\theta}_x \end{aligned} \quad (4.2.17)$$

and

$$\begin{aligned} a_{y2,2} - \frac{1}{2}\beta_{y2,2} \cos \tilde{\theta}_y &= \frac{r_{2,2}}{2\Delta y^2} (\kappa_{2,3} + \kappa_{2,1}) - \frac{r_{2,2}}{4\Delta y^2} \left((\kappa_{2,3} + \kappa_{2,1}) + \frac{\kappa_{2,3} - \kappa_{2,1}}{2} \right) \cos \tilde{\theta}_y \\ &= \frac{r_{2,2}}{2\Delta y^2} (\kappa_{2,3} + \kappa_{2,1}) - \frac{r_{2,2}}{4\Delta y^2} \left(\frac{3}{2}\kappa_{2,3} + \frac{1}{2}\kappa_{2,1} \right) \cos \tilde{\theta}_y, \end{aligned} \quad (4.2.18)$$

it can be seen that the (4.2.17) and (4.2.18) are greater or equal to 0 for $-1 \leq \cos \tilde{\theta}_x \leq 1$ and $-1 \leq \cos \tilde{\theta}_y \leq 1$. This is due to the spatial averaging operator $\square_{\tilde{\xi}}$ being introduced that prevent the difference in thermal conductivity, κ from becoming too high, therefore annihilate the effect of the gradient terms which causes instability. Using a similar process, the von Neumann method for other grid points can be obtained and shown to be stable. This substantiates that our proposed DG- and PR-ADI methods with the introduction of spatial averaging operator $\square_{\tilde{\xi}}$ are stable.

To justify the instability of the conventional PR-ADI method in Section 4.2, (4.2.17) and (4.2.18) are rewritten without the spatial operator $\square_{\tilde{\xi}}$ as

$$\begin{aligned} a_{x2,2} - \frac{1}{2}\beta_{x2,2} \cos \tilde{\theta}_x &= \frac{r_{2,2}}{\Delta x^2} \kappa_{2,2} - \frac{r_{2,2}}{2\Delta x^2} \left(\kappa_{2,2} + \frac{\kappa_{3,2} - \kappa_{1,2}}{4} \right) \cos \tilde{\theta}_x \\ &= \frac{r_{2,2}}{\Delta x^2} \kappa_{2,2} - \frac{r_{2,2}}{2\Delta x^2} \left(\kappa_{2,2} + \frac{1}{4}\kappa_{3,2} - \frac{1}{4}\kappa_{1,2} \right) \cos \tilde{\theta}_x \end{aligned} \quad (4.2.19)$$

and

$$\begin{aligned} a_{y2,2} - \frac{1}{2}\beta_{y2,2} \cos \tilde{\theta}_y &= \frac{r_{2,2}}{\Delta y^2} \kappa_{2,2} - \frac{r_{2,2}}{2\Delta y^2} \left(\kappa_{2,2} + \frac{\kappa_{2,3} - \kappa_{2,1}}{4} \right) \cos \tilde{\theta}_y \\ &= \frac{r_{2,2}}{\Delta y^2} \kappa_{2,2} - \frac{r_{2,2}}{2\Delta y^2} \left(\kappa_{2,2} + \frac{1}{4}\kappa_{2,3} - \frac{1}{4}\kappa_{2,1} \right) \cos \tilde{\theta}_y. \end{aligned} \quad (4.2.20)$$

There exists a value of κ , $\cos \tilde{\theta}_x$ and $\cos \tilde{\theta}_y$ such that (4.2.19) or (4.2.20) will be less than 0. For instance, if $\kappa_{3,2}$ is a very large number and $\cos \tilde{\theta}_x = 1$, (4.2.19) will be less than 0. Similarly, if $\kappa_{2,3}$ is a very large number and $\cos \tilde{\theta}_y = 1$, (4.2.20) will be less than 0. For both instances, the amplification matrix in (4.2.16) will be greater than 1, therefore resulting in the potential instability for the conventional DG- and

PR-ADI methods.

4.2.2 Stable and Efficient FADI Method

By comparing (4.2.8) and (4.2.13), it can be seen that the RHS of (4.2.13) is less complicated. However, it still has a considerable number of arithmetic operations. This is due to the operators that are found on the RHS of (4.2.11). To maximize efficiency, we rewrite (4.2.11) as

$$\tilde{V}_T|_{i,j}^n = \left(1 + \frac{1}{2}B_T\right)T_{PR}|_{i,j}^n \quad (4.2.21a)$$

$$\left(1 - \frac{1}{2}A_T\right)T_{PR}|_{i,j}^{n+\frac{1}{2}} = \tilde{V}_T|_{i,j}^n + \frac{1}{2}G|_{i,j}^n \quad (4.2.21b)$$

$$\tilde{V}_T|_{i,j}^{n+\frac{1}{2}} = \left(1 + \frac{1}{2}A_T\right)T_{PR}|_{i,j}^{n+\frac{1}{2}} + \frac{1}{2}G|_{i,j}^n \quad (4.2.21c)$$

$$\left(1 - \frac{1}{2}B_T\right)T_{PR}|_{i,j}^{n+1} = \tilde{V}_T|_{i,j}^{n+\frac{1}{2}} \quad (4.2.21d)$$

where $\tilde{V}_T$'s serve as auxiliary variables.

Next, we exploit the auxiliary variables in order to turn the algorithm above into a simpler one. In particular, based on (4.2.21d) at one time step backward

$$\tilde{V}_T|_{i,j}^{n-\frac{1}{2}} = \left(1 - \frac{1}{2}B_T\right)T_{PR}|_{i,j}^n \quad (4.2.22)$$

it follows that $\tilde{V}_T|_{i,j}^n$ of (4.2.21a) is reducible to

$$\begin{aligned} \tilde{V}_T|_{i,j}^n &= \left(1 + \frac{1}{2}B_T\right)T_{PR}|_{i,j}^n \\ &= 2T_{PR}|_{i,j}^n - \left(1 - \frac{1}{2}B_T\right)T_{PR}|_{i,j}^n \\ &= 2T_{PR}|_{i,j}^n - \tilde{V}_T|_{i,j}^{n-\frac{1}{2}}. \end{aligned} \quad (4.2.23)$$

Furthermore, upon recognizing (4.2.21b), $\tilde{V}_T|_{i,j}^{n+\frac{1}{2}}$ of (4.2.21c) is also reducible to

$$\begin{aligned} \tilde{V}_T|_{i,j}^{n+\frac{1}{2}} &= \left(1 + \frac{1}{2}A_T\right)T_{PR}|_{i,j}^{n+\frac{1}{2}} + \frac{1}{2}G|_{i,j}^n \\ &= 2T_{PR}|_{i,j}^{n+\frac{1}{2}} - \left(1 - \frac{1}{2}A_T\right)T_{PR}|_{i,j}^{n+\frac{1}{2}} + \frac{1}{2}G|_{i,j}^n \end{aligned}$$

$$= 2T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} - \tilde{V}_{\text{T}}|_{i,j}^{n+\frac{1}{2}}. \quad (4.2.24)$$

Note that G is no longer required in (4.2.24), thus the FADI method requires only single heat generation input in the first procedure [140]. With (4.2.23) and (4.2.24), algorithm (4.2.21) becomes

$$\tilde{V}_{\text{T}}|_{i,j}^n = 2T_{\text{PR}}|_{i,j}^n - \tilde{V}_{\text{T}}|_{i,j}^{n-\frac{1}{2}} \quad (4.2.25a)$$

$$\left(1 - \frac{1}{2}A_{\text{T}}\right)T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} = \tilde{V}_{\text{T}}|_{i,j}^n + \frac{1}{2}G|_{i,j}^n \quad (4.2.25b)$$

$$\tilde{V}_{\text{T}}|_{i,j}^{n+\frac{1}{2}} = 2T_{\text{PR}}|_{i,j}^{n+\frac{1}{2}} - \tilde{V}_{\text{T}}|_{i,j}^n \quad (4.2.25c)$$

$$\left(1 - \frac{1}{2}B_{\text{T}}\right)T_{\text{PR}}|_{i,j}^{n+1} = \tilde{V}_{\text{T}}|_{i,j}^{n+\frac{1}{2}}. \quad (4.2.25d)$$

Through re-definition of field variables

$$V_{\text{T}}|_{i,j}^{n-\frac{1}{2}} = \frac{1}{2}\tilde{V}_{\text{T}}|_{i,j}^{n-\frac{1}{2}}, \quad V_{\text{T}}|_{i,j}^n = \frac{1}{2}\tilde{V}_{\text{T}}|_{i,j}^n, \quad V_{\text{T}}|_{i,j}^{n+\frac{1}{2}} = \frac{1}{2}\tilde{V}_{\text{T}}|_{i,j}^{n+\frac{1}{2}},$$

we obtain the final update procedures as

$$V_{\text{T}}|_{i,j}^n = T_{\text{F}}|_{i,j}^n - V_{\text{T}}|_{i,j}^{n-\frac{1}{2}} \quad (4.2.26a)$$

$$\left(\frac{1}{2} - \frac{1}{4}A_{\text{T}}\right)T_{\text{F}}|_{i,j}^{n+\frac{1}{2}} = V_{\text{T}}|_{i,j}^n + \frac{1}{4}G|_{i,j}^n \quad (4.2.26b)$$

$$V_{\text{T}}|_{i,j}^{n+\frac{1}{2}} = T_{\text{F}}|_{i,j}^{n+\frac{1}{2}} - V_{\text{T}}|_{i,j}^n \quad (4.2.26c)$$

$$\left(\frac{1}{2} - \frac{1}{4}B_{\text{T}}\right)T_{\text{F}}|_{i,j}^{n+1} = V_{\text{T}}|_{i,j}^{n+\frac{1}{2}} \quad (4.2.26d)$$

with initialization

$$V_{\text{T}}|_{i,j}^{-\frac{1}{2}} = \left(\frac{1}{2} - \frac{1}{4}B_{\text{T}}\right)T_{\text{F}}|_{i,j}^0. \quad (4.2.27)$$

In this chapter, we shall also refer (4.2.26) as efficient fundamental ADI, or in short, FADI method to aptly describe such ADI method in the fundamental (basic) and simple form that cannot be reduced further on the RHS. There is neither (spatial) operator A_{T} nor B_{T} to be omitted or simplified on the RHS. It is worth noting that the FADI method for heat transfer in (4.2.26) here has its RHS free of scalar

operator A_T and B_T . This is different from the FADI method for electromagnetics in (3.2.10), which has its RHS free of matrix operators $\mathbf{A}_E$ and $\mathbf{B}_E$.

The temperatures resulted from both FADI and PR-ADI methods are identical, i.e. they have the relation

$$T_F|_{i,j}^n = T_{PR}|_{i,j}^n \quad (4.2.28a)$$

$$T_F|_{i,j}^{n+\frac{1}{2}} = T_{PR}|_{i,j}^{n+\frac{1}{2}} \quad (4.2.28b)$$

$$T_F|_{i,j}^{n+1} = T_{PR}|_{i,j}^{n+1}. \quad (4.2.28c)$$

Furthermore, the temperature acquired from the FADI method can be related to that of the DG-ADI method by

$$T_{DG}|_{i,j}^n = T_F|_{i,j}^n \quad (4.2.29a)$$

$$T_{DG}|_{i,j}^{n+\frac{1}{2}} = 2T_F|_{i,j}^{n+\frac{1}{2}} - T_F|_{i,j}^n \quad (4.2.29b)$$

$$T_{DG}|_{i,j}^{n+1} = T_F|_{i,j}^{n+1}. \quad (4.2.29c)$$

By applying central approximation and upon some manipulations for (4.2.26), we have

For first procedure from n to $n + \frac{1}{2}$:

$$V_T|_{i,j}^n = T_F|_{i,j}^n - V_T|_{i,j}^{n-\frac{1}{2}} \quad (4.2.30a)$$

$$-\frac{1}{4}\alpha_{x_{i,j}}T_F|_{i-1,j}^{n+\frac{1}{2}} + \frac{1}{2}(1 + a_{x_{i,j}})T_F|_{i,j}^{n+\frac{1}{2}} - \frac{1}{4}\beta_{x_{i,j}}T_F|_{i+1,j}^{n+\frac{1}{2}} = V_T|_{i,j}^n + \frac{1}{4}G_{i,j}^n \quad (4.2.30b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$V_T|_{i,j}^{n+\frac{1}{2}} = T_F|_{i,j}^{n+\frac{1}{2}} - V_T|_{i,j}^n \quad (4.2.30c)$$

$$-\frac{1}{4}\alpha_{y_{i,j}}T_F|_{i,j-1}^{n+1} + \frac{1}{2}(1 + a_{y_{i,j}})T_F|_{i,j}^{n+1} - \frac{1}{4}\beta_{y_{i,j}}T_F|_{i,j+1}^{n+1} = V_T|_{i,j}^{n+\frac{1}{2}}. \quad (4.2.30d)$$

For non-zero initial $T_F|_{i,j}^0$, we need to apply the initialization as follows:

$$V_T|_{i,j}^{-\frac{1}{2}} = -\frac{1}{4}\alpha_{y_{i,j}}T_F|_{i,j-1}^0 + \frac{1}{2}(1 + a_{y_{i,j}})T_F|_{i,j}^0 - \frac{1}{4}\beta_{y_{i,j}}T_F|_{i,j+1}^0. \quad (4.2.31)$$

For the DG-ADI method, the total flops count of the update equation on the RHS in (4.2.8) is 34 (16 A/S and 18 M/D). On the other hand, the PR-ADI method has a total flops count of 20 (8 A/S and 12 M/D) for the update equation on its RHS in (4.2.13) whereas the FADI method has a total flops count of 4 (3 A/S and 1 M/D) for the update equation on its RHS in (4.2.30). It can be seen that there is a reduction in total flops count of 16 when the FADI method is compared to the PR-ADI method. When the FADI method is further compared to the DG-ADI method, there is a greater reduction in total flops count of 30. As the overall flops count decreases, the overall efficiency increases.

4.2.3 Stability Analysis, Efficiency and Numerical Results

Stability Analysis

We first demonstrate the potential instability of the conventional DG-ADI method by considering a heat simulation layout of a piecewise homogeneous medium with area of $16 \text{ mm} \times 16 \text{ mm}$ as shown in Figure 4.1. The materials and associated parameters used in the piecewise homogeneous medium are as follows: silicon (thermal conductivity $\kappa = 131 \text{ W/m}\cdot\text{K}$, density $\rho = 2500 \text{ kg/m}^3$, specific heat $C_p = 700 \text{ J/kg}\cdot\text{K}$), alumina (thermal conductivity $\kappa = 20 \text{ W/m}\cdot\text{K}$, density $\rho = 2699 \text{ kg/m}^3$, specific heat $C_p = 901 \text{ J/kg}\cdot\text{K}$) and copper (thermal conductivity $\kappa = 401 \text{ W/m}\cdot\text{K}$, density $\rho = 8920 \text{ kg/m}^3$, specific heat $C_p = 385 \text{ J/kg}\cdot\text{K}$). The domain is discretized with spatial steps $\Delta x = \Delta y = 0.125 \text{ mm}$ resulting in a computation domain size of 129×129 grids. This computation domain is bounded by the convection boundary condition [54] in (2.2.2) (where $f(\vec{r}, t) = h_{\xi}T_{\infty}$) and surrounded by a heat bath with

Figure 4.1: Heat transfer simulation layout of a piecewise homogeneous medium.

an ambient temperature of T_∞ . By discretizing (2.2.2) at various boundaries ($i = 0$, $i = ie$, $j = 0$ and $j = je$), temperature variable T at $T|_{-1,j}, T|_{ie+1,j}, T|_{i,-1}, T|_{i,je+1}$ (virtual points which are out of the computation domain) at various time steps can be expressed as

$$T|_{-1,j}^n = T|_{1,j}^n + \frac{2\Delta x}{\kappa_{0,j}}(h_x^- T_\infty - h_x^- T|_{0,j}^n) \quad (4.2.32a)$$

$$T|_{ie+1,j}^n = T|_{ie-1,j}^n + \frac{2\Delta x}{\kappa_{ie,j}}(h_x^+ T_\infty - h_x^+ T|_{ie,j}^n) \quad (4.2.32b)$$

$$T|_{i,-1}^n = T|_{i,1}^n + \frac{2\Delta y}{\kappa_{i,0}}(h_y^- T_\infty - h_y^- T|_{i,0}^n) \quad (4.2.32c)$$

$$T|_{i,je+1}^n = T|_{i,je-1}^n + \frac{2\Delta y}{\kappa_{i,je}}(h_y^+ T_\infty - h_y^+ T|_{i,je}^n) \quad (4.2.32d)$$

where h_x^- , h_x^+ , h_y^- and h_y^+ are the effective heat transfer coefficients calculated from the equivalent thermal resistance on boundaries $x = 0$, $x = ie$, $y = 0$ and $y = je$, respectively. The effective heat transfer coefficients h_x^- , h_x^+ , h_y^- and h_y^+ for convection boundary conditions corresponding to left, right, bottom and top of the

Figure 4.2: Transient temperature at observation point ($i = 60, j = 60$) computed using the conventional DG-ADI method [c.f. (4.2.4)], $\gamma = 5$.

Figure 4.3: Transient temperature at observation point ($i = 60, j = 60$) computed using the FADI method [c.f. (4.2.30)] for various γ .

domain are set as $8.333 \times 10^3 \text{ W/m}^2 \cdot \text{K}$. Ambient temperature, T_∞ is assumed to be 26.85°C . The numerical simulation is performed for 3000 time steps and time

step Δt is specified as $\gamma = 5$ in (4.2.2). The heat energy generation rates (W/m^3) in alumina and copper are given by

$$g|_{i,j}^n = \begin{cases} \left(1 + \cos(2\pi f_0 t + \pi)\right) \times 10^9 & \text{if } 0 \leq t \leq \frac{1}{f_0} \\ 0 & \text{otherwise} \end{cases} \quad (4.2.33)$$

where $f_0 = 11.96$ Hz.

Figure 4.2 shows the transient temperature at observation point ($i = 60, j = 60$) computed using the conventional DG-ADI method [c.f. (4.2.4)]. The instability of the conventional DG-ADI method is evident as the transient temperature goes unbounded over time. On the other hand, Figure 4.3 shows the transient temperature at the same observation point, but computed using the proposed FADI method [c.f. (4.2.30)] for various γ . Through Figure 4.3, the temperature of the piecewise homogeneous medium has reached steady state. It can be seen that the transient temperature remains bounded and stable over time. The result computed using the explicit method is also included for comparison. It is observed that the results computed by the FADI method agree well with that of the explicit method, which substantiates our proposed method.

To further ascertain the potential instability of the conventional DG-ADI method, we resort to finding the eigenvalues of the amplification matrix of the system. Due to inhomogeneity of the domain, the traditional von-Neumann Fourier method cannot be utilized but instead, the actual discretized spatial-temporal equations have to be considered. However, it is noted that the size of the amplification matrix for the whole computation domain is very huge ($129^2 \times 129^2 = 16641 \times 16641$). To simplify the problem, we consider a smaller area of the domain bounded by $9.5 \text{ mm} \leq x \leq 11 \text{ mm}$ and $9.5 \text{ mm} \leq y \leq 11 \text{ mm}$, which results in a reduced amplification matrix with size $13^2 \times 13^2 = 169 \times 169$. Since the instability is caused by constraint (4.2.3), the area is chosen such that there is a change in thermal conductivity, κ

Figure 4.4: Scatter plot of eigenvalues of the reduced amplification matrix for the conventional DG-ADI method [c.f. (4.2.4)] for various γ .

in both x - and y -directions ($|\frac{\partial \kappa_{i,j}}{\partial x}| > 0$ and $|\frac{\partial \kappa_{i,j}}{\partial y}| > 0$). Figure 4.4 shows the scatter plot of eigenvalues of the reduced amplification matrix for the conventional DG-ADI method [c.f. (4.2.4)] for various γ . It can be seen that some eigenvalues are located outside the unit semi circle, indicating instability for the conventional DG-ADI method. Therefore, the transient temperature that goes unbounded over time as shown in Figure 4.2 is due to the potentially unstable conventional DG-ADI method and not late time instability. On the contrary, the scatter plot of eigenvalues for the FADI method [c.f. (4.2.30)] is shown in Figure 4.5. It can be observed that at larger γ values of 50 and 100, the method still maintains stability, with all eigenvalues confined within the unit semi circle.

Figure 4.5: Scatter plot of eigenvalues of the reduced amplification matrix for the FADI method [c.f. (4.2.30)] for various γ .

Efficiency

To justify the high efficiency gains achievable for the FADI method, we conduct numerical experiments and obtain the CPU computation time of DG-ADI, PR-ADI and FADI methods for a range of computation domains from 500×500 to 4000×4000 grids. The numerical simulation is performed for 3000 time steps with $\gamma = 5$. The programs are compiled using Microsoft Visual C++ under Microsoft Windows 7 operating system (OS) running on Intel Dual Core 2.66 GHz processor platform.

Table 4.1 shows the CPU efficiency gains of the FADI method over PR-ADI and DG-ADI methods for various computation domains. In particular, the efficiency gain of the FADI method over PR-ADI method increases from 1.4 to 4.3 as the domain

Table 4.1: Efficiency gains of the FADI method over PR-ADI and DG-ADI methods for various computation domains ($\gamma = 5$, 3000 time steps)

Domain Size	Efficiency Gain	
	FADI vs PR-ADI	FADI vs DG-ADI
500×500	1.438	2.344
1000×1000	1.339	2.258
1500×1500	1.422	2.505
2000×2000	1.422	2.592
2500×2500	1.696	3.880
3000×3000	2.154	6.179
3500×3500	3.053	7.145
4000×4000	4.339	7.687

increases. When the FADI method is further compared to the DG-ADI method, the efficiency gain would make a significant increase from 2.3 to 7.7! It can be seen that the DG-ADI method is the slowest among all three methods, due to its complicated RHS involving large amount of arithmetic operations, c.f. (4.2.8). The PR-ADI method is faster compared to the DG-ADI method as its RHS is less complicated, and requires less arithmetic operations, c.f. (4.2.13). The FADI method is the fastest among all methods due to its operator-free RHS; leading to a minimum number of arithmetic operations, c.f. (4.2.30).

Numerical Results

To ascertain the accuracy, we now consider an analytical example from [141] whereby the solution to the heat transfer equation of (2.2.1) subjected to boundary condition of (2.2.2) is predetermined as

$$T = \sin(2\pi v_t t) \cdot \sin(2\pi v_x x) \cdot \sin(2\pi v_y y) \quad (4.2.34)$$

Figure 4.6: Transient temperature at observation point $(i = 19, j = 21)$, computed by (a) the conventional DG-ADI method [c.f. (4.2.4)] (unstable) and (b) the FADI method [c.f. (4.2.30)] (stable).

where

$$v_t = 1, \quad v_x = 4, \quad v_y = 3.$$

The domain is bounded by $0 \leq x \leq 1$ m and $0 \leq y \leq 1$ m. In this example, κ is chosen as a continuous function given by

$$\kappa(x, y) = 1.01 + \cos(30\pi x) \cdot \cos(20\pi y)$$

Figure 4.7: Scatter plot of eigenvalues of the reduced amplification matrix for (a) conventional DG-ADI method [c.f. (4.2.4)] and (b) FADI method [c.f. (4.2.30)].

while constants $C_p = 1$, $\rho = 1$ and $h_{\tilde{\xi}} = 0$. Sources g and f are determined such that (2.2.1) and (2.2.2) are satisfied.

The above solution is reproduced numerically using the conventional DG-ADI and FADI methods. The domain is discretized with spatial steps $\Delta x = \Delta y = 0.02$ m and time step Δt is chosen as $\gamma = 100$. Figure 4.6 plots the transient temperature at observation point $(i = 19, j = 21)$, computed by (a) conventional DG-ADI method [c.f. (4.2.4)] and (b) FADI method [c.f. (4.2.30)]. The analytical solution given in (4.2.34) is also plotted in Figure 4.6b as reference. It is observed that for the conventional DG-ADI method, the computed solution has grown unbounded over time, once again exhibiting instability. On the other hand, the solution computed

by our FADI method remains stable and is in good agreement with the analytical solution.

We again perform an independent test by finding the eigenvalues of the reduced amplification matrix. The reduced area is bounded by $0.28 \text{ m} \leq x \leq 0.52 \text{ m}$ and $0.28 \text{ m} \leq y \leq 0.52 \text{ m}$, giving the reduced amplification matrix size of $13^2 \times 13^2 = 169 \times 169$. Figure 4.7 shows the scatter plot of eigenvalues of the reduced amplification matrix for (a) conventional DG-ADI method [c.f. (4.2.4)] and (b) FADI method [c.f. (4.2.30)]. The conventional DG-ADI method again exhibits instability, evident from its eigenvalues located outside the unit semi circle, whereas the eigenvalues of the FADI method are still bounded within the unit semi circle. These results further substantiate the stability and accuracy of our FADI method.

4.3 2-D ADI Method For ICs with microchannel cooling

In this section, we extend the 2-D FADI method for efficient thermal simulation of ICs with microchannel cooling. Unlike the FADI method with Laplacian and gradient terms in Section 4.2.2, the heat transfer equation is discretized with gradient terms being absorbed into the finite-difference operator directly without being expanded out. It is then incorporated into the GPU through CUDA implementation. Numerical results of efficiency gain and the cooling effect of the microchannels will be presented.

4.3.1 FADI method for ICs with microchannel cooling

For thermal simulation of ICs with microchannel cooling, the convection heat flux, $\rho(\vec{r})C_p(\vec{r})\vec{u}T(\vec{r}, t)$ due to fluid motion [142] needs to be included in the heat transfer

equation in (2.2.1) as follows:

$$\begin{aligned} \rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r},t)}{\partial t} &= \nabla \cdot [\kappa(\vec{r})\nabla T(\vec{r},t)] \\ &\quad - \rho(\vec{r})C_p(\vec{r})\nabla \cdot [\vec{u} T(\vec{r},t)] + g(\vec{r},t) \end{aligned} \quad (4.3.1)$$

where $\vec{u}$ is the fluid velocity vector (m/s).

Using the proposed FADI method, (4.3.1) can be formulated into

$$V_T|_{i,j}^n = T_F|_{i,j}^n - V_T|_{i,j}^{n-\frac{1}{2}} \quad (4.3.2a)$$

$$\left(\frac{1}{2} - \frac{1}{4}A_T\right)T_F|_{i,j}^{n+\frac{1}{2}} = V_T|_{i,j}^n + \frac{1}{4}G|_{i,j}^n \quad (4.3.2b)$$

$$V_T|_{i,j}^{n+\frac{1}{2}} = T_F|_{i,j}^{n+\frac{1}{2}} - V_T|_{i,j}^n \quad (4.3.2c)$$

$$\left(\frac{1}{2} - \frac{1}{4}B_T\right)T_F|_{i,j}^{n+1} = V_T|_{i,j}^{n+\frac{1}{2}} \quad (4.3.2d)$$

with initialization

$$V_T|_{i,j}^{-\frac{1}{2}} = \left(\frac{1}{2} - \frac{1}{4}B_T\right)T_F|_{i,j}^0 \quad (4.3.3)$$

where

$$\begin{aligned} A_T T|_{i,j} &= \frac{\Delta t}{\rho_{i,j}C_{p_{i,j}}} \left[\frac{\partial}{\partial x} \left(\kappa_{i,j} \frac{\partial T|_{i,j}}{\partial x} \right) - \rho_{i,j}C_{p_{i,j}} \frac{\partial}{\partial x} (u_x T|_{i,j}) \right], \\ B_T T|_{i,j} &= \frac{\Delta t}{\rho_{i,j}C_{p_{i,j}}} \left[\frac{\partial}{\partial y} \left(\kappa_{i,j} \frac{\partial T|_{i,j}}{\partial y} \right) - \rho_{i,j}C_{p_{i,j}} \frac{\partial}{\partial y} (u_y T|_{i,j}) \right]. \end{aligned}$$

and V_T 's serve as temporary auxiliary variables. It can be seen that (4.3.2) is similar to (4.2.26) in Section 4.2.2. However, (4.3.2) contain (spatial) operators $A_T T|_{i,j}$ and $B_T T|_{i,j}$ which involves the discretization of the heat transfer equation in (4.3.1) whereby the gradient terms are being absorbed into the finite-difference operator directly without being expanded out.

By applying central approximation and upon some manipulations for (4.3.2), we obtain

For first procedure from n to $n + \frac{1}{2}$:

$$V_T|_{i,j}^n = T_F|_{i,j}^n - V_T|_{i,j}^{n-\frac{1}{2}} \quad (4.3.4a)$$

$$-\frac{1}{4}\alpha_{x_{i,j}}T_F|_{i-1,j}^{n+\frac{1}{2}} + \frac{1}{2}(1 + a_{x_{i,j}})T_F|_{i,j}^{n+\frac{1}{2}} - \frac{1}{4}\beta_{x_{i,j}}T_F|_{i+1,j}^{n+\frac{1}{2}} = V_T|_{i,j}^n + \frac{1}{4}G|_{i,j}^n \quad (4.3.4b)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$V_T|_{i,j}^{n+\frac{1}{2}} = T_F|_{i,j}^{n+\frac{1}{2}} - V_T|_{i,j}^n \quad (4.3.4c)$$

$$-\frac{1}{4}\alpha_{y_{i,j}}T_F|_{i,j-1}^{n+1} + \frac{1}{2}(1 + a_{y_{i,j}})T_F|_{i,j}^{n+1} - \frac{1}{4}\beta_{y_{i,j}}T_F|_{i,j+1}^{n+1} = V_T|_{i,j}^{n+\frac{1}{2}}. \quad (4.3.4d)$$

where

$$\begin{aligned} u_x^+ &= \max\left(\frac{\rho_{i,j}C_{p_{i,j}}u_x}{\Delta x}, 0\right), & u_x^- &= \min\left(\frac{\rho_{i,j}C_{p_{i,j}}u_x}{\Delta x}, 0\right), \\ u_y^+ &= \max\left(\frac{\rho_{i,j}C_{p_{i,j}}u_y}{\Delta y}, 0\right), & u_y^- &= \min\left(\frac{\rho_{i,j}C_{p_{i,j}}u_y}{\Delta y}, 0\right), \\ \alpha_{x_{i,j}} &= r_{i,j}\left(\frac{\kappa_{i,j} + \kappa_{i-1,j}}{2\Delta x^2} + u_x^+\right), & \beta_{x_{i,j}} &= r_{i,j}\left(\frac{\kappa_{i+1,j} + \kappa_{i,j}}{2\Delta x^2} - u_x^-\right), \\ \alpha_{y_{i,j}} &= r_{i,j}\left(\frac{\kappa_{i,j} + \kappa_{i,j-1}}{2\Delta y^2} + u_y^+\right), & \beta_{y_{i,j}} &= r_{i,j}\left(\frac{\kappa_{i,j+1} + \kappa_{i,j}}{2\Delta y^2} - u_y^-\right), \\ a_{x_{i,j}} &= \frac{\alpha_{x_{i,j}} + \beta_{x_{i,j}}}{2}, & a_{y_{i,j}} &= \frac{\alpha_{y_{i,j}} + \beta_{y_{i,j}}}{2}. \end{aligned}$$

For non-zero initial $T_F|_{i,j}^0$, we need to apply the initialization as follows:

$$V_T|_{i,j}^{-\frac{1}{2}} = -\frac{1}{4}\alpha_{y_{i,j}}T_F|_{i,j-1}^0 + \frac{1}{2}(1 + a_{y_{i,j}})T_F|_{i,j}^0 - \frac{1}{4}\beta_{y_{i,j}}T_F|_{i,j+1}^0. \quad (4.3.5)$$

By comparing (4.2.30) and (4.3.4), it can be seen that the update coefficients are different. In (4.3.4), the update coefficients contain the convection heat flux which is discretized using the upwind scheme. On the other hand, (4.2.30) contains update coefficients of the discretized Laplacian and gradient terms.

Figure 4.8: Flowchart of the GPU-accelerated FADI method for thermal simulation.

4.3.2 GPU-accelerated FADI method for ICs with microchannel cooling

The FADI method is very attractive for its better conciseness, efficiency and programming simplicity. To further accelerate the FADI method, the GPU can be utilized through the implementation of CUDA [17–20, 98–100].

Figure 4.8 shows the flowchart of the GPU-accelerated FADI method for thermal simulation, where ie and je represent the number of cells in x - and y -directions respectively. In the first procedure, three kernels need to be invoked. The first kernel involves the explicit update of (4.3.2a) where data can be parallelized in both x - and y -directions. For (4.3.2b), it is an implicit update equation involving tridiagonal matrix on the LHS. Inverting tridiagonal matrix in (4.3.2b) involves the LU factorization where forward elimination is first required, followed by backward substitution, both along the x -direction (operator A_T). Therefore, the second kernel corresponds to forward elimination while the third kernel corresponds to backward substitution. It should be noted that since both forward elimination and backward substitution are performed along the x -direction, data can be parallelized in the y -direction. In the second procedure, the same kernel is invoked for (4.3.2c) as in (4.3.2a), while the forward elimination and backward substitution kernels for (4.3.2d) are performed along the y -direction (operator B_T). Hence, data can be

Table 4.2: Efficiency gains of the (GPU) FADI method over (CPU) PR-ADI, DG-ADI and explicit methods for various computation domains ($\gamma = 50$, 3×10^4 time steps)

Domain Size	Efficiency Gain		
	FADI vs PR-ADI	FADI vs DG-ADI	FADI vs Explicit
500×500	1.726	2.763	77.976
1000×1000	2.623	4.425	107.88
1500×1500	2.732	4.789	101.25
2000×2000	2.670	4.899	94.428
2500×2500	3.286	7.485	102.37
3000×3000	5.03	12.104	103.02
3500×3500	6.562	14.586	127.49

parallelized in the x -direction.

We conduct numerical experiments and obtain the GPU time of the FADI method for a range of computation domains from 500×500 to 3500×3500 grids. The numerical simulation is performed for 3×10^4 time steps with $\gamma = 50$. The GPU time of the FADI method is compared against the CPU time of DG-ADI, PR-ADI and explicit methods. Table 4.2 shows the efficiency gain of the FADI method over DG-ADI, PR-ADI and explicit methods for various computation domain. The efficiency gain for the FADI method over PR-ADI method increases approximately from 2 to 6 times. If the FADI method is compared to the DG-ADI method, the efficiency gain can increase up to 14 times. Lastly, the efficiency gain of the FADI method over explicit method ranges from 77 to 127 times. It is thus ascertained that substantial gain in the overall efficiency is achievable for the FADI method with GPU implementation.

Figure 4.9: Five-layer silicon structure.

4.3.3 Numerical Results

For numerical illustration, we consider a five-layer silicon structure of dimension $9.3 \text{ mm} \times 9.3 \text{ mm} \times 1.2 \text{ mm}$ with microchannel cooling shown in Figure 4.9. The functional blocks layer has thermal conductivity $\kappa = 315.82 \text{ W/m}\cdot\text{K}$, density $\rho = 8954.5 \text{ kg/m}^3$, specific heat $C_p = 383.65 \text{ J/kg}\cdot\text{K}$ and its power density is shown in Figure 4.10. For the microchannels layer, it consists of four microchannels along the x -direction with fluid velocity u_x , channel width and height set as 1.19 m/s , $700 \mu\text{m}$ and $300 \mu\text{m}$ [136], respectively. The fluid is water with thermal conductivity $\kappa = 0.613 \text{ W/m}\cdot\text{K}$, density $\rho = 997 \text{ kg/m}^3$ and specific heat $C_p = 4179 \text{ J/kg}\cdot\text{K}$. These two layers are separated by a silicon layer with thermal conductivity $\kappa = 131 \text{ W/m}\cdot\text{K}$, density $\rho = 2329 \text{ kg/m}^3$ and specific heat $C_p = 700 \text{ J/kg}\cdot\text{K}$. The top, bottom layers and all four side walls are terminated by convection boundary conditions in (4.2.32) and surrounded by a heat bath with an ambient temperature of T_∞ set as 26.85°C . For the four side walls, the effective heat transfer coefficients of h_x^- , h_x^+ , h_y^- and h_y^+ are $5 \times 10^3 \text{ W/m}^2\cdot\text{K}$. For the bottom and top layers, by using similar discretization of (4.2.32) at boundaries $k = 0$, $k = ke$, the effective heat transfer coefficients of h_z^- , h_z^+ are $6 \times 10^3 \text{ W/m}^2\cdot\text{K}$ and $7 \times 10^3 \text{ W/m}^2\cdot\text{K}$ respectively. Note that h_z^- , h_z^+ are calculated from the equivalent thermal resistance on boundaries of $k = 0$ and $k = ke$, respectively.

Figure 4.10: Power density of the functional blocks layer.

Figure 4.11: Temperature profile of the functional blocks layer at 0.2 s.

Figure 4.12: Transient temperature results at observation point ($i = 170$, $j = 120$, $k = 4$) with various γ .

To include the transfer of thermal energy across multi-layers, we use the change in heat flux in the vertical z -direction as part of the heat generation rate g . The specific heat flux q in the z -direction is given by

$$q = \kappa(z) \frac{\partial T(z, t)}{\partial z} - \rho(z) C_p(z) u_z T(z, t). \quad (4.3.6)$$

The heat generation rate g can then be obtained from the change of heat flux as

$$g = \frac{\partial q}{\partial z} = \frac{\partial}{\partial z} \left(\kappa(z) \frac{\partial T(z, t)}{\partial z} \right) - \rho(z) C_p(z) \frac{\partial}{\partial z} (u_z T(z, t)). \quad (4.3.7)$$

Using this approach, the domain is discretized with spatial steps $\Delta x = \Delta y = 39.9 \mu\text{m}$, while the vertical height between layers is set as $\Delta z = 300 \mu\text{m}$.

Figure 4.11 shows the temperature profile of the functional blocks layer at 0.2 s. The highest temperature is about 83°C . Taking an observation point at $i = 170$, $j = 120$,

Figure 4.13: Transient temperature results at observation point ($i = 170$, $j = 120$, $k = 4$) from zero to eight microchannels.

$k = 4$ on the structure, the transient temperature results for various γ are shown in Figure 4.12. It can be seen that the temperature of the five-layer silicon structure has reached steady state and the FADI method is stable with no oscillations for $\gamma > 0.5$. The results are compared against the explicit method and they agree well.

To investigate the effect of the number of microchannels on temperature, we further plot the transient temperature at observation point ($i = 170$, $j = 120$, $k = 4$) with zero to eight microchannels in Figure 4.13. It can be observed that as the number of microchannels increases, the steady state temperature decreases. The time taken to reach steady state also decreases. Furthermore, it is worth noting that the structure experiences the most significant drop in temperature when the number of microchannels is added from zero to two, which further ascertains its cooling effect.

4.4 3-D ADI Method

In this section, we discuss the derivation of the conventional 3-D DG-ADI method into 3-D efficient ADI method for homogeneous media. This is followed by the comparison in terms of memory allocation and efficiency gain for both efficient ADI and DG-ADI methods.

4.4.1 Efficient ADI method

Douglas-Gunn [60] and Peaceman-Rachford [61] developed two variations of the ADI methods which are applicable to the 3-D thermal problems. However, among these two methods, the PR-ADI method is only conditionally stable. Therefore, in this section, we consider only the DG-ADI method.

According to (2.2.9), we can rearrange the DG-ADI method into the compact form as

$$\left(1 - \frac{1}{2}\hat{A}_T\right)T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} = \left(1 + \frac{1}{2}\hat{A}_T + \hat{B}_T + \hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \quad (4.4.1a)$$

$$\left(1 - \frac{1}{2}\hat{B}_T\right)T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} = \left(1 + \frac{1}{2}\hat{A}_T + \frac{1}{2}\hat{B}_T + \hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n + \frac{1}{2}\hat{A}_T T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + G|_{i,j,k}^n \quad (4.4.1b)$$

$$\begin{aligned} \left(1 - \frac{1}{2}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^{n+1} &= \left(1 + \frac{1}{2}\hat{A}_T + \frac{1}{2}\hat{B}_T + \frac{1}{2}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n + \frac{1}{2}\hat{A}_T T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} \\ &\quad + \frac{1}{2}\hat{B}_T T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + G|_{i,j,k}^n \end{aligned} \quad (4.4.1c)$$

where

$$\hat{A}_T = \hat{r}_{i,j,k} \frac{\partial^2}{\partial x^2}, \quad \hat{B}_T = \hat{r}_{i,j,k} \frac{\partial^2}{\partial y^2}, \quad \hat{C}_T = \hat{r}_{i,j,k} \frac{\partial^2}{\partial z^2}.$$

There are operators $\hat{A}_T$, $\hat{B}_T$ and $\hat{C}_T$ on the RHS for all three procedures in the update equations of (4.4.1). It leads to higher complexity in implementation with substantial amount of arithmetic operations as shown in (2.2.10). To simplify the

implementation which will lead to a reduction in the amount of arithmetic operations, we modify (4.4.1) into

$$\left(\frac{1}{2} - \frac{1}{4}\hat{A}_T\right)T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} = \left(\frac{1}{2} + \frac{1}{4}\hat{B}_T + \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n \quad (4.4.2a)$$

$$\left(\frac{1}{2} - \frac{1}{4}\hat{B}_T\right)T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} = \left(\frac{1}{4}\hat{B}_T + \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n + \frac{1}{4}\hat{A}_T T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{4}G|_{i,j,k}^n \quad (4.4.2b)$$

$$\begin{aligned} \left(\frac{1}{4} - \frac{1}{8}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^{n+1} &= \left(\frac{1}{2} + \frac{1}{4}\hat{B}_T + \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n - \frac{1}{2}\left(\frac{1}{2} + \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n \\ &\quad + \frac{1}{4}\hat{A}_T T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{4}\hat{B}_T T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{4}G|_{i,j,k}^n \end{aligned} \quad (4.4.2c)$$

where

$$\begin{aligned} T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} &= \frac{1}{2}\left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n\right), \\ T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} &= \frac{1}{2}\left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} - T_{\text{DG}}|_{i,j,k}^n\right). \end{aligned}$$

By substituting (4.4.2a) into (4.4.2b) and (4.4.2c), we get

$$\left(1 - \frac{1}{2}\hat{B}_T\right)T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} = T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} - T_{\text{DG}}|_{i,j,k}^n \quad (4.4.3a)$$

$$\left(\frac{1}{2} - \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^{n+1} = T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{2}\hat{B}_T T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} - \left(\frac{1}{2} + \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n \quad (4.4.3b)$$

Subsequently, we substitute (4.4.3a) into (4.4.3b) to obtain

$$\left(\frac{1}{2} - \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^{n+1} = T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} + \left(\frac{1}{2} - \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n \quad (4.4.4)$$

We can now rewrite (4.4.2) with (4.4.3a) and (4.4.4) as

$$\left(\frac{1}{2} - \frac{1}{4}\hat{A}_T\right)T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} = \left(\frac{1}{2} + \frac{1}{4}\hat{B}_T + \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n \quad (4.4.5a)$$

$$\left(1 - \frac{1}{2}\hat{B}_T\right)T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} = T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} - T_{\text{DG}}|_{i,j,k}^n \quad (4.4.5b)$$

$$V_T|_{i,j,k}^{n+1} = T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} + \left(\frac{1}{2} - \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^n \quad (4.4.5c)$$

$$\left(\frac{1}{2} - \frac{1}{4}\hat{C}_T\right)T_{\text{DG}}|_{i,j,k}^{n+1} = V_T|_{i,j,k}^{n+1} \quad (4.4.5d)$$

where V_T 's serve as auxiliary variables. To further reduce the operators involved

in (4.4.5), we exploit the auxiliary variables based on (4.4.5d) at one time step backward

$$V_T|_{i,j,k}^n = \left(\frac{1}{2} - \frac{1}{4}\widehat{C}_T\right)T_{DG}|_{i,j,k}^n \quad (4.4.6)$$

it follows that $V_T|_{i,j,k}^{n+\frac{1}{3}}$ of (4.4.5a) is reducible to

$$\begin{aligned} \left(\frac{1}{2} - \frac{1}{4}\widehat{A}_T\right)T_{EFF}|_{i,j,k}^{n+\frac{1}{3}} &= \left(\frac{1}{2} + \frac{1}{4}\widehat{B}_T + \frac{1}{4}\widehat{C}_T\right)T_{DG}|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n \\ &= \left(1 + \frac{1}{4}\widehat{B}_T\right)T_{DG}|_{i,j,k}^n - \left(\frac{1}{2} - \frac{1}{4}\widehat{C}_T\right)T_{DG}|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n \\ &= \left(1 + \frac{1}{4}\widehat{B}_T\right)T_{DG}|_{i,j,k}^n - V_T|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n. \end{aligned} \quad (4.4.7)$$

Furthermore, $V_T|_{i,j,k}^{n+1}$ of (4.4.5c) is also reducible to

$$\begin{aligned} V_T|_{i,j,k}^{n+1} &= T_{EFF}|_{i,j,k}^{n+\frac{2}{3}} + \left(\frac{1}{2} - \frac{1}{4}\widehat{C}_T\right)T_{DG}|_{i,j,k}^n \\ &= T_{EFF}|_{i,j,k}^{n+\frac{2}{3}} + V_T|_{i,j,k}^n. \end{aligned} \quad (4.4.8)$$

With (4.4.7) and (4.4.8), algorithm (4.4.5) becomes

$$\left(\frac{1}{2} - \frac{1}{4}\widehat{A}_T\right)T_{EFF}|_{i,j,k}^{n+\frac{1}{3}} = \left(1 + \frac{1}{4}\widehat{B}_T\right)T_{EFF}|_{i,j,k}^n - V_T|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n \quad (4.4.9a)$$

$$\left(1 - \frac{1}{2}\widehat{B}_T\right)T_{EFF}|_{i,j,k}^{n+\frac{2}{3}} = T_{EFF}|_{i,j,k}^{n+\frac{1}{3}} - T_{EFF}|_{i,j,k}^n \quad (4.4.9b)$$

$$V_T|_{i,j,k}^{n+1} = T_{EFF}|_{i,j,k}^{n+\frac{2}{3}} + V_T|_{i,j,k}^n \quad (4.4.9c)$$

$$\left(\frac{1}{2} - \frac{1}{4}\widehat{C}_T\right)T_{EFF}|_{i,j,k}^{n+1} = V_T|_{i,j,k}^{n+1} \quad (4.4.9d)$$

with initialization

$$V_T|_{i,j,k}^0 = \left(\frac{1}{2} - \frac{1}{4}\widehat{C}_T\right)T_{EFF}|_{i,j,k}^0 \quad (4.4.10)$$

It is worth noting that G is no longer required in (4.4.9b)-(4.4.9d), thus the method requires only single heat generation input in the first procedure.

The temperatures resulted from both efficient ADI and DG-ADI methods are simi-

lar, i.e. they have the relation

$$T_{\text{DG}}|_{i,j,k}^n = T_{\text{EFF}}|_{i,j,k}^n \quad (4.4.11)$$

$$T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} = 2T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} - T_{\text{EFF}}|_{i,j,k}^n \quad (4.4.12)$$

$$T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} = 2T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{EFF}}|_{i,j,k}^n \quad (4.4.13)$$

$$T_{\text{DG}}|_{i,j,k}^{n+1} = T_{\text{EFF}}|_{i,j,k}^{n+1} \quad (4.4.14)$$

It can be seen that the efficient ADI method contains only (spatial) operator $\hat{B}_T$ in the first procedure of the compact form. There is no more (spatial) operator $\hat{A}_T$ or $\hat{C}_T$ to be omitted or simplified on the RHS. On the other hand, the DG-ADI method still contains operators $\hat{A}_T$, $\hat{B}_T$ and $\hat{C}_T$ for all three procedures on the RHS pending further reduction.

By applying central approximation and arithmetic manipulation for (4.4.9), we have

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} & \frac{1}{2}(1 + \hat{r}_{x_{i,j,k}})T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} - \frac{1}{4}\hat{r}_{x_{i,j,k}} \left(T_{\text{EFF}}|_{i-1,j,k}^{n+\frac{1}{3}} + T_{\text{EFF}}|_{i+1,j,k}^{n+\frac{1}{3}} \right) \\ &= \frac{1}{4}\hat{r}_{y_{i,j,k}} \left(T_{\text{EFF}}|_{i,j-1,k}^n + T_{\text{EFF}}|_{i,j+1,k}^n \right) + (1 - \frac{1}{2}\hat{r}_{y_{i,j,k}})T_{\text{EFF}}|_{i,j,k}^n - V_T|_{i,j,k}^n + \frac{1}{4}G|_{i,j,k}^n \end{aligned} \quad (4.4.15a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$(1 + \hat{r}_{y_{i,j,k}})T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} - \frac{1}{2}\hat{r}_{y_{i,j,k}} \left(T_{\text{EFF}}|_{i,j-1,k}^{n+\frac{2}{3}} + T_{\text{EFF}}|_{i,j+1,k}^{n+\frac{2}{3}} \right) = T_{\text{EFF}}|_{i,j,k}^{n+\frac{1}{3}} - T_{\text{EFF}}|_{i,j,k}^n \quad (4.4.15b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$V_T|_{i,j,k}^{n+1} = T_{\text{EFF}}|_{i,j,k}^{n+\frac{2}{3}} + V_T|_{i,j,k}^n \quad (4.4.15c)$$

$$\frac{1}{2}(1 + \hat{r}_{z_{i,j,k}})T_{\text{EFF}}|_{i,j,k}^{n+1} - \frac{1}{4}\hat{r}_{z_{i,j,k}} \left(T_{\text{EFF}}|_{i,j,k-1}^{n+1} + T_{\text{EFF}}|_{i,j,k+1}^{n+1} \right) = V_T|_{i,j,k}^{n+1} \quad (4.4.15d)$$

Initialization of $\mathbf{v}$	// $\mathbf{v} \leftarrow V_T^0$ in (4.4.10)
for $n = 0, 1, 2, \dots$	// $n \leftarrow$ time step in the main iteration
$\mathbf{u} = (1 + \frac{1}{4}\widehat{B}_T)\mathbf{t} - \mathbf{v}$	// $\mathbf{t} \leftarrow T_{\text{EFF}}^n$, $\mathbf{v} \leftarrow V_T^n$, $\mathbf{u} \leftarrow$ RHS of (4.4.9a)
$\mathbf{u} = \text{inv}(\frac{1}{2} - \frac{1}{4}\widehat{A}_T)\mathbf{u}$	// $\mathbf{u}$ at LHS $\leftarrow T_{\text{EFF}}^{n+\frac{1}{3}}$, $\mathbf{u}$ at RHS $\leftarrow$ RHS of (4.4.9a)
$\mathbf{u} = \mathbf{u} - \mathbf{t}$	// $\mathbf{t} \leftarrow T_{\text{EFF}}^n$, $\mathbf{u}$ at RHS $\leftarrow T_{\text{EFF}}^{n+\frac{1}{3}}$, $\mathbf{u}$ at LHS $\leftarrow$ RHS of (4.4.9b)
$\mathbf{u} = \text{inv}(1 - \frac{1}{2}\widehat{B}_T)\mathbf{u}$	// $\mathbf{u}$ at LHS $\leftarrow T_{\text{EFF}}^{n+\frac{2}{3}}$, $\mathbf{u}$ at RHS $\leftarrow$ RHS of (4.4.9b)
$\mathbf{v} = \mathbf{u} - \mathbf{v}$	// $\mathbf{u} \leftarrow T_{\text{EFF}}^{n+\frac{2}{3}}$, $\mathbf{v}$ at RHS $\leftarrow V_T^n$, $\mathbf{v}$ at LHS $\leftarrow V_T^{n+1}$
$\mathbf{t} = \text{inv}(\frac{1}{2} - \frac{1}{4}\widehat{C}_T)\mathbf{v}$	// $\mathbf{t} \leftarrow T_{\text{EFF}}^{n+1}$, $\mathbf{v} \leftarrow V_T^{n+1}$
end	

Figure 4.14: Pseudocode of 3-D efficient ADI method across iterations.

For non-zero initial $T_{\text{EFF}}|_{i,j,k}^0$, we apply the initialization as follows:

$$V_T|_{i,j,k}^0 = \frac{1}{2}(1 + \widehat{r}_{z_{i,j,k}})T_{\text{EFF}}|_{i,j,k}^0 - \frac{1}{4}\widehat{r}_{z_{i,j,k}}(T_{\text{EFF}}|_{i,j,k+1}^0 + T_{\text{EFF}}|_{i,j,k-1}^0) \quad (4.4.16)$$

By comparing (2.2.10) and (4.4.15), it is apparent that the update equations of (4.4.15) are definitely more concise and simple, leading to a reduction in the amount of arithmetic operations.

4.4.2 Memory Allocation

The reuse of memory space is important for reducing the frequency of invoking virtual memory. To increase the efficiency of the thermal simulation, it is of great importance to minimize the amount of memory usage.

Figure 4.14 presents the pseudocode of 3-D efficient ADI method across iterations. It can be seen that three variables, i.e. $\mathbf{t}$, $\mathbf{u}$ and $\mathbf{v}$ are required. The memory spaces are reused where $\mathbf{t}$ is multi-purpose and may represent T_{EFF}^n , T_{EFF}^{n+1} . $\mathbf{v}$ is multi-purpose and may represent V_T^n and V_T^{n+1} . $\mathbf{u}$ is multi-purpose and may represent $T_{\text{EFF}}^{n+\frac{1}{3}}$, $T_{\text{EFF}}^{n+\frac{2}{3}}$, intermediate variables for the RHS of (4.4.9a) and (4.4.9b). On the other hand, the pseudocode of 3-D DG-ADI method in Figure 4.15 requires four

for $n = 0, 1, 2, \dots$	// $n \leftarrow$ time step in the main iteration
$\mathbf{w} = (1 + \frac{1}{2}\hat{A}_T + \hat{B}_T + \hat{C}_T)\mathbf{t}$	// $\mathbf{t} \leftarrow T_{\text{DG}}^n$, $\mathbf{w} \leftarrow$ RHS of (4.4.1a)
$\mathbf{v} = \text{inv}(1 - \frac{1}{2}\hat{A}_T)\mathbf{w}$	// $\mathbf{v} \leftarrow T_{\text{DG}}^{n+\frac{1}{3}}$, $\mathbf{w} \leftarrow$ RHS of (4.4.1a)
$\mathbf{w} = (1 + \frac{1}{2}\hat{A}_T + \frac{1}{2}\hat{B}_T + \hat{C}_T)\mathbf{t} + \frac{1}{2}\hat{A}_T\mathbf{v}$	// $\mathbf{t} \leftarrow T_{\text{DG}}^n$, $\mathbf{v} \leftarrow T_{\text{DG}}^{n+\frac{1}{3}}$, $\mathbf{w} \leftarrow$ RHS of (4.4.1b)
$\mathbf{u} = \text{inv}(1 - \frac{1}{2}\hat{B}_T)\mathbf{w}$	// $\mathbf{u} \leftarrow T_{\text{DG}}^{n+\frac{2}{3}}$, $\mathbf{w} \leftarrow$ RHS of (4.4.1b)
$\mathbf{w} = (1 + \frac{1}{2}\hat{A}_T + \frac{1}{2}\hat{B}_T + \frac{1}{2}\hat{C}_T)\mathbf{t}$ $\quad + \frac{1}{2}\hat{A}_T\mathbf{v} + \frac{1}{2}\hat{B}_T\mathbf{u}$	// $\mathbf{t} \leftarrow T_{\text{DG}}^n$, $\mathbf{v} \leftarrow T_{\text{DG}}^{n+\frac{1}{3}}$, $\mathbf{u} \leftarrow T_{\text{DG}}^{n+\frac{2}{3}}$, // $\mathbf{w} \leftarrow$ RHS of (4.4.1c)
$\mathbf{t} = \text{inv}(1 - \frac{1}{2}\hat{C}_T)\mathbf{w}$	// $\mathbf{t} \leftarrow T_{\text{DG}}^{n+1}$, $\mathbf{w} \leftarrow$ RHS of (4.4.1c)
end	

Figure 4.15: Pseudocode of 3-D DG-ADI method across iterations.

variables, i.e. $\mathbf{t}$, $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ for computation. There is an additional one more variable $\mathbf{w}$ that is required for the DG-ADI method as compared to the efficient ADI method. Therefore, the efficient ADI method is more appealing than the DG-ADI method due to the reduction in the required amount of memory variables. This in turn reduces the memory space and the memory indexing overhead.

4.4.3 Boundary Conditions

The pseudocodes in Figures 4.14 and 4.15 present the memory allocation and general flow of simulated program. However, there exists another issue for the implicit methods at the boundary of the computation domain.

For example, a simulation domain of $ie \times je \times ke$ grids, where the grids indexed from 0 to ie in the x -direction, 0 to je in the y -direction and 0 to ke in the z -direction. Temperature variable T at $T_{-1,j,k}$, $T_{ie+1,j,k}$, $T_{i,-1,k}$, $T_{i,je+1,k}$, $T_{i,j,-1}$ or $T_{i,j,ke+1}$ (virtual points) will be out of the simulation domain. These virtual points will result in extra efforts for the implementation of tridiagonal system of equations.

In order to cater for general thermal simulation, we treat the boundary [54] by discretizing (2.2.2) at various boundaries ($i = 0$, $i = ie$, $j = 0$, $j = je$, $k = 0$,


```

First Procedure
// Main grid
for j=1:je-1
for k=1:ke-1
    i=0;
    Eq. (2.2.10a) with BCs for  $T_{\text{DG}}|_{-1,j,k}^{n+\frac{1}{3}}$  &  $T_{\text{DG}}|_{-1,j,k}^n$ 
    for i=1:ie-1
        Eq. (2.2.10a)
    end
    i=ie;
    Eq. (2.2.10a) with BCs for  $T_{\text{DG}}|_{ie+1,j,k}^{n+\frac{1}{3}}$  &  $T_{\text{DG}}|_{ie+1,j,k}^n$ 
end
end
end

```

Figure 4.16: Pseudocode of 3-D DG-ADI method in main grid for the first procedure of iteration.

$k = ke$), the virtual points at various time steps can be expressed as

$$T|_{-1,j,k}^n = T|_{1,j,k}^n + \frac{2\Delta x}{\kappa_{0,j,k}}(h_x^- T_\infty - h_x^- T|_{0,j,k}^n) \quad (4.4.17a)$$

$$T|_{ie+1,j,k}^n = T|_{ie-1,j,k}^n + \frac{2\Delta x}{\kappa_{ie,j,k}}(h_x^+ T_\infty - h_x^+ T|_{ie,j,k}^n) \quad (4.4.17b)$$

$$T|_{i,-1,k}^n = T|_{i,1,k}^n + \frac{2\Delta y}{\kappa_{i,0,k}}(h_y^- T_\infty - h_y^- T|_{i,0,k}^n) \quad (4.4.17c)$$

$$T|_{i,je+1,k}^n = T|_{i,je-1,k}^n + \frac{2\Delta y}{\kappa_{i,je,k}}(h_y^+ T_\infty - h_y^+ T|_{i,je,k}^n) \quad (4.4.17d)$$

$$T|_{i,j,-1}^n = T|_{i,j,1}^n + \frac{2\Delta z}{\kappa_{i,j,0}}(h_z^- T_\infty - h_z^- T|_{i,j,0}^n) \quad (4.4.17e)$$

$$T|_{i,j,ke+1}^n = T|_{i,j,ke-1}^n + \frac{2\Delta z}{\kappa_{i,j,ke}}(h_z^+ T_\infty - h_z^+ T|_{i,j,ke}^n) \quad (4.4.17f)$$

where h_x^- , h_x^+ , h_y^- , h_y^+ , h_z^- and h_z^+ are the effective heat transfer coefficients calculated from the equivalent thermal resistance on boundaries $x = 0$, $x = ie$, $y = 0$, $y = je$, $z = 0$ and $z = ke$, respectively.

Figure 4.16 shows the pseudocode of 3-D DG-ADI method in the main grid for the

```

First Procedure
// Boundary j=0
j=0;
for k=1:ke-1
    i=0;
    Eq. (2.2.10a) with BCs for  $T_{DG}|_{-1,0,k}^{n+\frac{1}{3}}$ ,  $T_{DG}|_{-1,0,k}^n$  &  $T_{DG}|_{0,-1,k}^n$ 
    for i=1:ie-1
        Eq. (2.2.10a) with BC for  $T_{DG}|_{i,-1,k}^n$ 
    end
    i=ie;
    Eq. (2.2.10a) with BCs for  $T_{DG}|_{ie+1,0,k}^{n+\frac{1}{3}}$ ,  $T_{DG}|_{ie+1,0,k}^n$  &  $T_{DG}|_{ie,-1,k}^n$ 
end
// Boundary j=0 & k=0
j=0; k=0;
i=0;
Eq. (2.2.10a) with BCs for  $T_{DG}|_{-1,0,0}^{n+\frac{1}{3}}$ ,  $T_{DG}|_{-1,0,0}^n$ ,  $T_{DG}|_{0,-1,0}^n$  &  $T_{DG}|_{0,0,-1}^n$ 
for i=1:ie-1
    Eq. (2.2.10a) with BCs for  $T_{DG}|_{i,-1,0}^n$  &  $T_{DG}|_{i,0,-1}^n$ 
end
i=ie;
Eq. (2.2.10a) with BCs for  $T_{DG}|_{ie+1,0,0}^{n+\frac{1}{3}}$ ,  $T_{DG}|_{ie+1,0,0}^n$ ,  $T_{DG}|_{ie,-1,0}^n$  &  $T_{DG}|_{ie,0,-1}^n$ 
// Boundary j=0 & k=ke
j=0; k=ke;
i=0;
Eq. (2.2.10a) with BCs for  $T_{DG}|_{-1,0,ke}^{n+\frac{1}{3}}$ ,  $T_{DG}|_{-1,0,ke}^n$ ,  $T_{DG}|_{0,-1,ke}^n$  &  $T_{DG}|_{0,0,ke+1}^n$ 
for i=1:ie-1
    Eq. (2.2.10a) with BCs for  $T_{DG}|_{i,-1,ke}^n$  &  $T_{DG}|_{i,0,ke+1}^n$ 
end
i=ie;
Eq. (2.2.10a) with BCs for  $T_{DG}|_{ie+1,0,ke}^{n+\frac{1}{3}}$ ,  $T_{DG}|_{ie+1,0,ke}^n$ ,  $T_{DG}|_{ie,-1,ke}^n$  &  $T_{DG}|_{ie,0,ke+1}^n$ 

```

Figure 4.17: Pseudocode of 3-D DG-ADI method in $x - z$ plane at $y = 0$ for the first procedure of iteration.

first procedure of iteration. The update equations in the main grid includes $y - z$ boundary planes. However, the DG-ADI method requires additional update equations for boundaries in $x - y$ and $x - z$ planes. Figure 4.17 shows the pseudocode of 3-D DG-ADI method in $x - z$ plane at $y = 0$ for first procedure of iteration. Note that update equations of other boundary planes can be written by permuting the indices. It can be seen that a great number of for-loops is required merely for the

```

First Procedure
// Main grid
for j=1:je-1
for k=0:ke
    i=0;
        Eq. (4.4.15a) with BC for  $T_{\text{EFF}}|_{-1,j,k}^{n+\frac{1}{3}}$ 
    for i=1:ie-1
        Eq. (4.4.15a)
    end
    i=ie;
        Eq. (4.4.15a) with BC for  $T_{\text{EFF}}|_{ie+1,j,k}^{n+\frac{1}{3}}$ 
    end
end
end

```

Figure 4.18: Pseudocode of the 3-D efficient ADI method in main grid for the first procedure of iteration.

first procedure. The second and third procedures would follow the same amount of additional for-loops for boundaries to be implemented. Due to the boundary condition (BC) that has to be implemented, many for-loops on boundaries are required for the DG-ADI method. In addition, there is a substantial number of virtual point treatments on boundaries in each update equation. This makes the implementation of the DG-ADI method complicated and computationally expensive.

Figure 4.18 shows the pseudocode of 3-D efficient ADI method in the main grid for the first procedure of iteration. The update equations in the main grid includes both $x - y$ and $y - z$ planes. Additional update equations for boundaries are only required for $x - z$ plane. Figure 4.19 shows the pseudocode of 3-D efficient ADI method in $x - z$ plane at $y = 0$ for the first procedure of iteration. Note that update equations for boundary plane $x - z$ plane at $y = je$ can be written by permuting the indices. It can be seen that there is a decrease in the number of for-loops required in the first procedure as compared to the DG-ADI method. Furthermore,

```

First Procedure
// Boundary j=0
j=0;
for k=0:ke
    i=0;
    Eq. (4.4.15a) with BCs for  $T_{\text{EFF}}|_{-1,0,k}^{n+\frac{1}{3}}$  &  $T_{\text{EFF}}|_{0,-1,k}^n$ 
    for i=1:ie-1
        Eq. (4.4.15a) with BC for  $T_{\text{EFF}}|_{i,-1,k}^n$ 
    end
    i=ie;
    Eq. (4.4.15a) with BCs for  $T_{\text{EFF}}|_{ie+1,0,k}^{n+\frac{1}{3}}$  &  $T_{\text{EFF}}|_{ie,-1,k}^n$ 
end

```

Figure 4.19: Pseudocode of 3-D efficient ADI method in $x - z$ plane at $y = 0$ for the first procedure of iteration.

the second and third procedures require only a single for-loop for the main grids as the boundary conditions can be implemented together. This further enhances the implementation of the efficient ADI method over DG-ADI method for the solving heat transfer equation.

4.4.4 Numerical Experiments

To compare and validate the efficient ADI method with DG-ADI method, we conduct numerical simulations of a chip with layout of its hierarchy function blocks and power density shown in Figure 4.20. The chip size is $1.5 \text{ mm} \times 2.4 \text{ mm} \times 0.3 \text{ mm}$ and the spatial steps are $\Delta x = \Delta y = \Delta z = 15 \mu\text{m}$. Ambient temperature, T_∞ is assumed to be 26.85°C . The convection boundary conditions on the sides of the chip have an effective heat transfer coefficients of $h_x^- = h_x^+ = h_y^- = h_y^+ = 3 \times 10^3 \text{ W/m}^2\cdot\text{K}$. The primary and secondary heat transfer paths have effective heat transfer coefficients of $h_z^+ = 2.5 \times 10^3 \text{ W/m}^2\cdot\text{K}$ and $h_z^- = 3.5 \times 10^4 \text{ W/m}^2\cdot\text{K}$ respectively.

Figure 4.20: A chip layout with the power density in each hierarchical function block.

Figure 4.21: Temperature profile of the chip at 0.12 s.

Figure 4.22: Transient temperature results using DG-ADI and efficient ADI methods with various γ .

Figure 4.21 shows the temperature profile of the chip at 0.12 s. The highest temperature is at 100 °C which is influenced by the effective heat transfer coefficients. Parameters that affect the temperature include board-level component population (thermal loading), heat sink style and design, and air velocity on the components and/or heat sink.

Figure 4.22 shows the transient temperature results using DG-ADI and efficient ADI methods with various γ at observation point $i = 50$, $j = 60$ and $k = 13$. Through Figure 4.22, the temperature of the chip has reached steady state. It can be seen that the temperatures of both DG-ADI and efficient ADI methods are identical, which in turn confirm that both methods are equivalent. In addition, these two ADI methods are unconditionally stable as there is no instability even for $\gamma > \frac{1}{2}$.

After verifying the efficient ADI method with DG-ADI method, we provide a comparison for the computation efficiency gain between the two methods. The numerical

Table 4.3: Efficiency gains of the efficient ADI method over DG-ADI method for various computation domain ($\gamma=2$, 3000 time steps)

Domain Size	Efficiency Gain
	Efficient ADI vs DG-ADI
$50 \times 50 \times 50$	3.52
$100 \times 100 \times 100$	3.29
$150 \times 150 \times 150$	3.14
$200 \times 200 \times 200$	3.13
$250 \times 250 \times 250$	3.19

simulation is performed for 3000 time steps with $\gamma = 2$. Note that the programs have been compiled using Microsoft Visual C++ under Microsoft Windows 7 operating system (OS) running on Intel Dual Core 2.66 GHz processor platform. The CPU computation time is obtained for both DG-ADI and efficient ADI methods for a range of computation domains from $50 \times 50 \times 50$ to $250 \times 250 \times 250$ grids.

Table 4.3 shows the CPU efficiency gains of the efficient ADI method over DG-ADI method for various computation domains. The DG-ADI method involves large amount of arithmetic operations, c.f. (4.4.1) due to the existence of operators $\hat{A}_T$, $\hat{B}_T$ and $\hat{C}_T$ on the RHS of all three procedures. On the other hand, the efficient ADI method involves a minimum number of arithmetic operations, c.f. (4.4.9) as the algorithm contains only operator $\hat{B}_T$ in the first procedure, with operator-free RHS in the second and third procedures. Thus, we ascertain the high efficiency gain (of up to 3.5 times) for the efficient ADI method over DG-ADI method.

4.5 3-D LOD Method

The previous section presented the formulation of 3-D efficient ADI method within homogeneous media for solving the heat transfer equation. However, by extending

the proposed stabilized 2-D DG-ADI method in Section 4.2.1 into 3-D DG-ADI method, the potential instability caused by the gradient terms within inhomogeneous media cannot be resolved. To overcome this, we present the two stable 3-D LOD methods for solving the heat transfer equation. The LOD method is then cast into the compact form and formulated into FLOD method with operator-free RHS. Stability analysis and the maximum relative error of the FLOD method will be investigated.

4.5.1 Potentially Unstable DG-ADI method with gradient terms

Extending the proposed stabilized DG-ADI method within inhomogeneous media into the 3-D case, the heat transfer equation in (2.2.8) can now be written as

$$T_{\text{DG}}|_{i,j,k}^{n+1} = r_{i,j,k} \left[\square_x \kappa_{i,j,k} \frac{\partial^2}{2\partial x^2} + \frac{\partial \kappa_{i,j,k}}{\partial x} \frac{\partial}{2\partial x} + \square_y \kappa_{i,j,k} \frac{\partial^2}{2\partial y^2} + \frac{\partial \kappa_{i,j,k}}{\partial y} \frac{\partial}{2\partial y} + \square_z \kappa_{i,j,k} \frac{\partial^2}{2\partial z^2} + \frac{\partial \kappa_{i,j,k}}{\partial z} \frac{\partial}{2\partial z} \right] (T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n) + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \quad (4.5.1)$$

where

$$r_{i,j,k} = \frac{\Delta t}{\rho_{i,j,k} C_{p_{i,j,k}}}, \quad \square_x \kappa_{i,j,k} = \frac{\kappa_{i+1,j,k} + \kappa_{i-1,j,k}}{2},$$

$$\square_y \kappa_{i,j,k} = \frac{\kappa_{i,j+1,k} + \kappa_{i,j-1,k}}{2}, \quad \square_z \kappa_{i,j,k} = \frac{\kappa_{i,j,k+1} + \kappa_{i,j,k-1}}{2}.$$

According to the DG-ADI method, the update equation in (4.5.1), inclusive of both Laplacian and gradient terms is written in three procedures as

For first procedure from n to $n + \frac{1}{3}$:

$$T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} = \frac{r_{i,j,k}}{2} \left[\square_x \kappa_{i,j,k} \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j,k}}{\partial x} \frac{\partial}{\partial x} \right] (T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n) + r_{i,j,k} \left[\square_y \kappa_{i,j,k} \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j,k}}{\partial y} \frac{\partial}{\partial y} \right] T_{\text{DG}}|_{i,j,k}^n$$

$$+r_{i,j,k} \left[\square_z \kappa_{i,j,k} \frac{\partial^2}{\partial z^2} + \frac{\partial \kappa_{i,j,k}}{\partial z} \frac{\partial}{\partial z} \right] T_{\text{DG}}|_{i,j,k}^n + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \quad (4.5.2a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} &= \frac{r_{i,j,k}}{2} \left[\square_x \kappa_{i,j,k} \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j,k}}{\partial x} \frac{\partial}{\partial x} \right] \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) \\ &+ \frac{r_{i,j,k}}{2} \left[\square_y \kappa_{i,j,k} \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j,k}}{\partial y} \frac{\partial}{\partial y} \right] \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) \\ &+ r_{i,j,k} \left[\square_z \kappa_{i,j,k} \frac{\partial^2}{\partial z^2} + \frac{\partial \kappa_{i,j,k}}{\partial z} \frac{\partial}{\partial z} \right] T_{\text{DG}}|_{i,j,k}^n + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \end{aligned} \quad (4.5.2b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned} T_{\text{DG}}|_{i,j,k}^{n+1} &= \frac{r_{i,j,k}}{2} \left[\square_x \kappa_{i,j,k} \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j,k}}{\partial x} \frac{\partial}{\partial x} \right] \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) \\ &+ \frac{r_{i,j,k}}{2} \left[\square_y \kappa_{i,j,k} \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j,k}}{\partial y} \frac{\partial}{\partial y} \right] \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j,k}^n \right) \\ &+ \frac{r_{i,j,k}}{2} \left[\square_z \kappa_{i,j,k} \frac{\partial^2}{\partial z^2} + \frac{\partial \kappa_{i,j,k}}{\partial z} \frac{\partial}{\partial z} \right] \left(T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n \right) + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \end{aligned} \quad (4.5.2c)$$

However, unlike 2-D ADI method, even with the inclusion of spatial averaging operator, the 3-D DG-ADI method is still conditionally stable. This may be due to the 3-D DG-ADI method update procedures [c.f. (4.5.2)] which are of very different form as compared to the 2-D ADI method update procedures [c.f. (4.2.11)]. To achieve an unconditionally stable implicit method for the heat transfer equation, we proposed 3-D LOD method which will be described below.

4.5.2 Proposed Stable LOD method with gradient terms

To formulate 3-D LOD method, we first rearrange (4.5.1) into the compact form as

$$\left(1 - \frac{1}{2} A_T \right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} = \left(1 + \frac{1}{2} A_T \right) T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3} G|_{i,j,k}^n \quad (4.5.3a)$$

$$\left(1 - \frac{1}{2} B_T \right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} = \left(1 + \frac{1}{2} B_T \right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{3} G|_{i,j,k}^n \quad (4.5.3b)$$

$$\left(1 - \frac{1}{2}C_T\right)T_{\text{LOD}}|_{i,j,k}^{n+1} = \left(1 + \frac{1}{2}C_T\right)T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{3}G|_{i,j,k}^n \quad (4.5.3c)$$

where

$$\begin{aligned} A_T &= r_{i,j,k} \left[\square_x \kappa_{i,j,k} \frac{\partial^2}{\partial x^2} + \frac{\partial \kappa_{i,j,k}}{\partial x} \frac{\partial}{\partial x} \right], \quad B_T = r_{i,j,k} \left[\square_y \kappa_{i,j,k} \frac{\partial^2}{\partial y^2} + \frac{\partial \kappa_{i,j,k}}{\partial y} \frac{\partial}{\partial y} \right], \\ C_T &= r_{i,j,k} \left[\square_z \kappa_{i,j,k} \frac{\partial^2}{\partial z^2} + \frac{\partial \kappa_{i,j,k}}{\partial z} \frac{\partial}{\partial z} \right]. \end{aligned}$$

Note that the variables A_T and B_T above are for 3-D LOD method, which are slightly different (in terms of indices) from (4.2.7). By comparing the proposed stabilized 2-D ADI method update procedures [c.f. (4.2.11)] with the 3-D LOD method update procedures [c.f. (4.5.3)], it can be seen that both methods have quite similar form. Therefore, the 2-D ADI method with spatial averaging operator can be extended into the 3-D LOD method to annihilate the effect of the gradient terms which cause instability.

By applying central approximation and arithmetic manipulation for (4.5.3), we have For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} & -\frac{1}{2}\alpha_{x_{i,j,k}}T_{\text{LOD}}|_{i-1,j,k}^{n+\frac{1}{3}} + (1 + a_{x_{i,j,k}})T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} - \frac{1}{2}\beta_{x_{i,j,k}}T_{\text{LOD}}|_{i+1,j,k}^{n+\frac{1}{3}} \\ & = \frac{1}{2}\alpha_{x_{i,j,k}}T_{\text{LOD}}|_{i-1,j,k}^n + \frac{1}{2}\beta_{x_{i,j,k}}T_{\text{LOD}}|_{i+1,j,k}^n + (1 - a_{x_{i,j,k}})T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G|_{i,j,k}^n \end{aligned} \quad (4.5.4a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} & -\frac{1}{2}\alpha_{y_{i,j,k}}T_{\text{LOD}}|_{i,j-1,k}^{n+\frac{2}{3}} + (1 + a_{y_{i,j,k}})T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} - \frac{1}{2}\beta_{y_{i,j,k}}T_{\text{LOD}}|_{i,j+1,k}^{n+\frac{2}{3}} \\ & = \frac{1}{2}\alpha_{y_{i,j,k}}T_{\text{LOD}}|_{i,j-1,k}^{n+\frac{1}{3}} + \frac{1}{2}\beta_{y_{i,j,k}}T_{\text{LOD}}|_{i,j+1,k}^{n+\frac{1}{3}} + (1 - a_{y_{i,j,k}})T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{3}G|_{i,j,k}^n \end{aligned} \quad (4.5.4b)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$-\frac{1}{2}\alpha_{z_{i,j,k}}T_{\text{LOD}}|_{i,j,k-1}^{n+1} + (1 + a_{z_{i,j,k}})T_{\text{LOD}}|_{i,j,k}^{n+1} - \frac{1}{2}\beta_{z_{i,j,k}}T_{\text{LOD}}|_{i,j,k+1}^{n+1}$$

$$= \frac{1}{2} \alpha_{z_{i,j,k}} T_{\text{LOD}}|_{i,j,k-1}^{n+\frac{2}{3}} + \frac{1}{2} \beta_{z_{i,j,k}} T_{\text{LOD}}|_{i,j,k+1}^{n+\frac{2}{3}} + (1 - a_{z_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{3} G|_{i,j,k}^n \quad (4.5.4c)$$

where

$$\begin{aligned} \alpha_{\xi_{i,j,k}} &= a_{\xi_{i,j,k}} - b_{\xi_{i,j,k}}, & \beta_{\xi_{i,j,k}} &= a_{\xi_{i,j,k}} + b_{\xi_{i,j,k}}, \\ a_{\xi_{i,j,k}} &= \frac{r_{i,j,k}}{\Delta \xi^2} \square_{\xi} \kappa_{i,j,k}, & b_{\xi_{i,j,k}} &= \frac{r_{i,j,k}}{2\Delta \xi} \frac{\partial \kappa_{i,j,k}}{\partial \xi}. \end{aligned}$$

The proposed 3-D LOD method here is an unconditionally stable implicit method which is able to circumvent the stability constraint for both Laplacian and gradient terms in the heat transfer equation. The spatial averaging operator is introduced to the Laplacian terms of the LOD method to annihilate the effect of gradient terms which causes instability.

To prove that the proposed 3-D LOD method is stable, we make use of the von Neumann method at every grid point of the computation domain. The temperature along the boundary of the domain is set as 0 K. For instance, by converting (4.5.4) at grid point $i = 2, j = 2, k = 2$ in Fourier domain and combining all three procedures, the amplification matrix for the LOD method can be written as

$$\begin{aligned} & \left| \frac{1 - \left[a_{x_{2,2,2}} - \frac{1}{2} \beta_{x_{2,2,2}} \cos \tilde{\theta}_x \right] - \mathbf{j} \frac{1}{2} \beta_{x_{2,2,2}} \sin \tilde{\theta}_x}{1 + \left[a_{x_{2,2,2}} - \frac{1}{2} \beta_{x_{2,2,2}} \cos \tilde{\theta}_x \right] - \mathbf{j} \frac{1}{2} \beta_{x_{2,2,2}} \sin \tilde{\theta}_x} \right. \\ & \quad \cdot \frac{1 - \left[a_{y_{2,2,2}} - \frac{1}{2} \beta_{y_{2,2,2}} \cos \tilde{\theta}_y \right] - \mathbf{j} \frac{1}{2} \beta_{y_{2,2,2}} \sin \tilde{\theta}_y}{1 + \left[a_{y_{2,2,2}} - \frac{1}{2} \beta_{y_{2,2,2}} \cos \tilde{\theta}_y \right] - \mathbf{j} \frac{1}{2} \beta_{y_{2,2,2}} \sin \tilde{\theta}_y} \\ & \quad \cdot \left. \frac{1 - \left[a_{z_{2,2,2}} - \frac{1}{2} \beta_{z_{2,2,2}} \cos \tilde{\theta}_z \right] - \mathbf{j} \frac{1}{2} \beta_{z_{2,2,2}} \sin \tilde{\theta}_z}{1 + \left[a_{z_{2,2,2}} - \frac{1}{2} \beta_{z_{2,2,2}} \cos \tilde{\theta}_z \right] - \mathbf{j} \frac{1}{2} \beta_{z_{2,2,2}} \sin \tilde{\theta}_z} \right| \leq 1 \quad (4.5.5) \end{aligned}$$

It can be seen that the amplification matrix of 3-D FLOD method in (4.5.5) is similar to 2-D PR-ADI method in (4.2.16) with an addition of the Fourier domain discretized update equation of (4.5.4c) in the z -direction. Using a similar technique, the von Neumann method for other grid points can be obtained and shown to be

stable. This ascertains that our proposed 3-D LOD method is stable.

4.5.3 Potentially Unstable DG-ADI method without gradient terms

By discretizing the heat transfer equation (2.2.1) whereby the gradient terms has been absorbed into the finite-difference operator directly without being expanded out, (2.2.1) for general inhomogeneous material can now be expanded into

$$\begin{aligned} \rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial t} &= \frac{\partial}{\partial x} \left(\kappa(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial x} \right) + \frac{\partial}{\partial y} \left(\kappa(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial y} \right) \\ &+ \frac{\partial}{\partial z} \left(\kappa(\vec{r})\frac{\partial T(\vec{r}, t)}{\partial z} \right) + g(\vec{r}, t) \end{aligned} \quad (4.5.6)$$

We first discretize (4.5.6) into

$$\begin{aligned} T_{\text{DG}}|_{i,j,k}^{n+1} &= r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n)}{2\partial x} \right) \\ &+ r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n)}{2\partial y} \right) + T_{\text{DG}}|_{i,j,k}^n \\ &+ r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n)}{2\partial z} \right) + G|_{i,j,k}^n \end{aligned} \quad (4.5.7)$$

According to the DG-ADI method, the update equation in (4.5.7) is solved in three procedures as

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} &= r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial (T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n)}{2\partial x} \right) \\ &+ r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial T_{\text{DG}}|_{i,j,k}^n}{\partial y} \right) \\ &+ r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial T_{\text{DG}}|_{i,j,k}^n}{\partial z} \right) + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n \end{aligned} \quad (4.5.8a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned}
T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} = & r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right)}{2\partial x} \right) \\
& + r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j,k}^n \right)}{2\partial y} \right) \\
& + r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial T_{\text{DG}}|_{i,j,k}^n}{\partial z} \right) + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n
\end{aligned} \tag{4.5.8b}$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned}
T_{\text{DG}}|_{i,j,k}^{n+1} = & r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{DG}}|_{i,j,k}^n \right)}{2\partial x} \right) \\
& + r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial \left(T_{\text{DG}}|_{i,j,k}^{n+\frac{2}{3}} + T_{\text{DG}}|_{i,j,k}^n \right)}{2\partial y} \right) \\
& + r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial \left(T_{\text{DG}}|_{i,j,k}^{n+1} + T_{\text{DG}}|_{i,j,k}^n \right)}{2\partial z} \right) + T_{\text{DG}}|_{i,j,k}^n + G|_{i,j,k}^n
\end{aligned} \tag{4.5.8c}$$

All spatial derivatives are approximated using central differencing. The 3-D DG-ADI method for the heat transfer equation within inhomogeneous media is still potentially unstable (to be demonstrated in Section 4.5.8). It is worth noting that the DG-ADI method with gradient terms being absorbed into the finite-difference operator directly without being expanded out become more stable (although still remain not unconditionally stable) as compared to (4.5.2). In order to achieve an unconditionally stable implicit method for the heat transfer equation, we now propose the 3-D LOD method which will be described in the next section.

4.5.4 Proposed Stable LOD method without gradient terms

Similar to Section 4.5.2, to formulate 3-D LOD method whereby the gradient terms has been absorbed into the finite-difference operator directly without being expanded

out, we rearrange (4.5.7) into the compact form as

$$\left(1 - \frac{1}{2}A_T\right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} = \left(1 + \frac{1}{2}A_T\right) T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G|_{i,j,k}^n \quad (4.5.9a)$$

$$\left(1 - \frac{1}{2}B_T\right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} = \left(1 + \frac{1}{2}B_T\right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{3}G|_{i,j,k}^n \quad (4.5.9b)$$

$$\left(1 - \frac{1}{2}C_T\right) T_{\text{LOD}}|_{i,j,k}^{n+1} = \left(1 + \frac{1}{2}C_T\right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{3}G|_{i,j,k}^n \quad (4.5.9c)$$

where

$$A_T T|_{i,j,k} = r_{i,j,k} \frac{\partial}{\partial x} \left(\kappa_{i,j,k} \frac{\partial T_{i,j,k}}{\partial x} \right), \quad B_T T|_{i,j,k} = r_{i,j,k} \frac{\partial}{\partial y} \left(\kappa_{i,j,k} \frac{\partial T_{i,j,k}}{\partial y} \right),$$

$$C_T T|_{i,j,k} = r_{i,j,k} \frac{\partial}{\partial z} \left(\kappa_{i,j,k} \frac{\partial T_{i,j,k}}{\partial z} \right).$$

Note that the operators above are different to that of Section 4.5.2. It can be seen that (4.5.9) contain (spatial) operators $A_T T|_{i,j,k}$, $B_T T|_{i,j,k}$ and $C_T T|_{i,j,k}$ which involves the discretization of the heat transfer equation whereby the gradient terms are being absorbed into the finite-difference operator directly without being expanded out. On the other hand, (4.5.3) contain (spatial) operators A_T , B_T and C_T which involves the discretization of the heat transfer equation with spatial averaging operator $\square_\xi$.

By applying central difference approximation for spatial operators in (4.5.9) and upon some arithmetic manipulations, we have

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} & -\frac{1}{2}\alpha_{x_{i,j,k}} T_{\text{LOD}}|_{i-1,j,k}^{n+\frac{1}{3}} + (1 + a_{x_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} - \frac{1}{2}\beta_{x_{i,j,k}} T_{\text{LOD}}|_{i+1,j,k}^{n+\frac{1}{3}} \\ & = \frac{1}{2}\alpha_{x_{i,j,k}} T_{\text{LOD}}|_{i-1,j,k}^n + \frac{1}{2}\beta_{x_{i,j,k}} T_{\text{LOD}}|_{i+1,j,k}^n \\ & + (1 - a_{x_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G|_{i,j,k}^n \end{aligned} \quad (4.5.10a)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$-\frac{1}{2}\alpha_{y_{i,j,k}} T_{\text{LOD}}|_{i,j-1,k}^{n+\frac{2}{3}} + (1 + a_{y_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} - \frac{1}{2}\beta_{y_{i,j,k}} T_{\text{LOD}}|_{i,j+1,k}^{n+\frac{2}{3}}$$

$$\begin{aligned}
&= \frac{1}{2} \alpha_{y_{i,j,k}} T_{\text{LOD}}|_{i,j-1,k}^{n+\frac{1}{3}} + \frac{1}{2} \beta_{y_{i,j,k}} T_{\text{LOD}}|_{i,j+1,k}^{n+\frac{1}{3}} \\
&+ (1 - a_{y_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{3} G|_{i,j,k}^n
\end{aligned} \tag{4.5.10b}$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned}
&-\frac{1}{2} \alpha_{z_{i,j,k}} T_{\text{LOD}}|_{i,j,k-1}^{n+1} + (1 + a_{z_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^{n+1} - \frac{1}{2} \beta_{z_{i,j,k}} T_{\text{LOD}}|_{i,j,k+1}^{n+1} \\
&= \frac{1}{2} \alpha_{z_{i,j,k}} T_{\text{LOD}}|_{i,j,k-1}^{n+\frac{2}{3}} + \frac{1}{2} \beta_{z_{i,j,k}} T_{\text{LOD}}|_{i,j,k+1}^{n+\frac{2}{3}} \\
&+ (1 - a_{z_{i,j,k}}) T_{\text{LOD}}|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{3} G|_{i,j,k}^n
\end{aligned} \tag{4.5.10c}$$

where

$$\begin{aligned}
\alpha_{x_{i,j,k}} &= r_{i,j,k} \frac{\kappa_{i,j,k} + \kappa_{i-1,j,k}}{2\Delta x^2}, & \beta_{x_{i,j,k}} &= r_{i,j,k} \frac{\kappa_{i+1,j,k} + \kappa_{i,j,k}}{2\Delta x^2}, \\
\alpha_{y_{i,j,k}} &= r_{i,j,k} \frac{\kappa_{i,j,k} + \kappa_{i,j-1,k}}{2\Delta y^2}, & \beta_{y_{i,j,k}} &= r_{i,j,k} \frac{\kappa_{i,j+1,k} + \kappa_{i,j,k}}{2\Delta y^2}, \\
\alpha_{z_{i,j,k}} &= r_{i,j,k} \frac{\kappa_{i,j,k} + \kappa_{i,j,k-1}}{2\Delta z^2}, & \beta_{z_{i,j,k}} &= r_{i,j,k} \frac{\kappa_{i,j,k+1} + \kappa_{i,j,k}}{2\Delta z^2}, \\
a_{x_{i,j,k}} &= \frac{\alpha_{x_{i,j,k}} + \beta_{x_{i,j,k}}}{2}, & a_{y_{i,j,k}} &= \frac{\alpha_{y_{i,j,k}} + \beta_{y_{i,j,k}}}{2}, & a_{z_{i,j,k}} &= \frac{\alpha_{z_{i,j,k}} + \beta_{z_{i,j,k}}}{2}.
\end{aligned}$$

Comparing (4.5.4) and (4.5.10), it can be seen that their update coefficients are different. In (4.5.4), the update coefficients contain $b_{\xi_{i,j,k}}$ and spatial averaging operator $\square_{\xi}$. These update coefficients are omitted in (4.5.10) due to the gradient terms being absorbed into the finite-difference operator without being expanded out.

4.5.5 Stable and Efficient FLOD Method

Despite having a single operator on the RHS of the updating equations, the LOD method in Section 4.5.4 can be further reduced into the efficient fundamental scheme similar to electromagnetics in Section 3.2.2. From (4.5.9a), we have

$$\begin{aligned}
\left(1 - \frac{1}{2} A_T\right) T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} &= \left(1 + \frac{1}{2} A_T\right) T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3} G|_{i,j,k}^n \\
&= 2T_{\text{LOD}}|_{i,j,k}^n - \left(1 - \frac{1}{2} A_T\right) T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3} G|_{i,j,k}^n.
\end{aligned} \tag{4.5.11}$$

This can be manipulated readily to give

$$\left(1 - \frac{1}{2}A_T\right) \left(T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{LOD}}|_{i,j,k}^n\right) = 2T_{\text{LOD}}|_{i,j,k}^n + \frac{1}{3}G|_{i,j,k}^n \quad (4.5.12)$$

where the scalar terms in bracket may be denoted by auxiliary variable

$$V_T|_{i,j,k}^{n+\frac{1}{3}} = T_{\text{LOD}}|_{i,j,k}^{n+\frac{1}{3}} + T_{\text{LOD}}|_{i,j,k}^n. \quad (4.5.13)$$

Similar manipulation applies to (4.5.9b) and (4.5.9c) which leads to auxiliary variables $V_T|_{i,j,k}^{n+\frac{2}{3}}$ and $V_T|_{i,j,k}^{n+1}$ respectively. Combining all auxiliary and field variables, we can obtain 3-D FLOD method written as

$$\left(\frac{1}{2} - \frac{1}{4}A_T\right) V_T|_{i,j,k}^{n+\frac{1}{3}} = T_F|_{i,j,k}^n + \frac{1}{6}G|_{i,j,k}^n \quad (4.5.14a)$$

$$T_F|_{i,j,k}^{n+\frac{1}{3}} = V_T|_{i,j,k}^{n+\frac{1}{3}} - T_F|_{i,j,k}^n \quad (4.5.14b)$$

$$\left(\frac{1}{2} - \frac{1}{4}B_T\right) V_T|_{i,j,k}^{n+\frac{2}{3}} = T_F|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{6}G|_{i,j,k}^n \quad (4.5.14c)$$

$$T_F|_{i,j,k}^{n+\frac{2}{3}} = V_T|_{i,j,k}^{n+\frac{2}{3}} - T_F|_{i,j,k}^{n+\frac{1}{3}} \quad (4.5.14d)$$

$$\left(\frac{1}{2} - \frac{1}{4}C_T\right) V_T|_{i,j,k}^{n+1} = T_F|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{6}G|_{i,j,k}^n \quad (4.5.14e)$$

$$T_F|_{i,j,k}^{n+1} = V_T|_{i,j,k}^{n+1} - T_F|_{i,j,k}^{n+\frac{2}{3}} \quad (4.5.14f)$$

This algorithm has no (spatial) operators on the RHS, hence it leads to a decrease in the overall flops count as well as an increase in the overall efficiency. Note that the FLOD method for electromagnetics in (3.2.24) is not directly applicable here because it involves two procedures with its RHS free of matrix operators $\mathbf{A}_E$ and $\mathbf{B}_E$. The FLOD method for heat transfer in (4.5.14) here involves three procedures with its RHS free from scalar operators A_T , B_T and C_T .

Both FLOD and LOD methods are one in the same. Therefore the temperature of the FLOD method for each time step, i.e. T_F^n , $T_F^{n+\frac{1}{3}}$, $T_F^{n+\frac{2}{3}}$ and T_F^{n+1} directly corresponds to those of the LOD method. The equivalence of both FLOD and LOD methods becomes evident here.

By applying central approximation and arithmetic manipulation for (4.5.14), we have

For first procedure from n to $n + \frac{1}{3}$:

$$\begin{aligned} -\frac{1}{4}\alpha_{x_{i,j,k}} V_T|_{i-1,j,k}^{n+\frac{1}{3}} + \frac{1}{2}(1 + a_{x_{i,j,k}}) V_T|_{i,j,k}^{n+\frac{1}{3}} - \frac{1}{4}\beta_{x_{i,j,k}} V_T|_{i+1,j,k}^{n+\frac{1}{3}} \\ = T_F|_{i,j,k}^n + \frac{1}{6}G|_{i,j,k}^n \end{aligned} \quad (4.5.15a)$$

$$T_F|_{i,j,k}^{n+\frac{1}{3}} = V_T|_{i,j,k}^{n+\frac{1}{3}} - T_F|_{i,j,k}^n \quad (4.5.15b)$$

For second procedure from $n + \frac{1}{3}$ to $n + \frac{2}{3}$:

$$\begin{aligned} -\frac{1}{4}\alpha_{y_{i,j,k}} V_T|_{i,j-1,k}^{n+\frac{2}{3}} + \frac{1}{2}(1 + a_{y_{i,j,k}}) V_T|_{i,j,k}^{n+\frac{2}{3}} - \frac{1}{4}\beta_{y_{i,j,k}} V_T|_{i,j+1,k}^{n+\frac{2}{3}} \\ = T_F|_{i,j,k}^{n+\frac{1}{3}} + \frac{1}{6}G|_{i,j,k}^n \end{aligned} \quad (4.5.15c)$$

$$T_F|_{i,j,k}^{n+\frac{2}{3}} = V_T|_{i,j,k}^{n+\frac{2}{3}} - T_F|_{i,j,k}^{n+\frac{1}{3}} \quad (4.5.15d)$$

For third procedure from $n + \frac{2}{3}$ to $n + 1$:

$$\begin{aligned} -\frac{1}{4}\alpha_{z_{i,j,k}} V_T|_{i,j,k-1}^{n+1} + \frac{1}{2}(1 + a_{z_{i,j,k}}) V_T|_{i,j,k}^{n+1} - \frac{1}{4}\beta_{z_{i,j,k}} V_T|_{i,j,k+1}^{n+1} \\ = T_F|_{i,j,k}^{n+\frac{2}{3}} + \frac{1}{6}G|_{i,j,k}^n \end{aligned} \quad (4.5.15e)$$

$$T_F|_{i,j,k}^{n+1} = V_T|_{i,j,k}^{n+1} - T_F|_{i,j,k}^{n+\frac{2}{3}} \quad (4.5.15f)$$

By comparing (4.5.10) and (4.5.15), we find that the update equations of (4.5.15) are obviously the most computationally efficient. This is due to the operator-free RHS of the FLOD algorithm. Note that although not shown here, using this technique, we are also able to reduce the FLOD method in (4.5.3) into the fundamental form.

4.5.6 Memory Allocation

At first sight, one might think that (4.5.15) incurs more memory resources due to the presence of auxiliary variables V_T 's. To clarify the memory allocation for various variables, the pseudocode of implicit methods across iterations is shown in

for $n = 0, 1, 2, \dots$	// $n \leftarrow$ time step in the main iteration
$\mathbf{v} = \text{inv}(\frac{1}{2} - \frac{1}{4}A)\mathbf{t}$	// $\mathbf{t} \leftarrow T_F^n, \mathbf{v} \leftarrow V_T^{n+\frac{1}{3}}$
$\mathbf{t} = \mathbf{v} - \mathbf{t}$	// $\mathbf{v} \leftarrow V_T^{n+\frac{1}{3}}, \mathbf{t}$ at RHS $\leftarrow T_F^n, \mathbf{t}$ at LHS $\leftarrow T_F^{n+\frac{1}{3}}$
$\mathbf{v} = \text{inv}(\frac{1}{2} - \frac{1}{4}B)\mathbf{t}$	// $\mathbf{t} \leftarrow T_F^{n+\frac{1}{3}}, \mathbf{v} \leftarrow V_T^{n+\frac{2}{3}}$
$\mathbf{t} = \mathbf{v} - \mathbf{t}$	// $\mathbf{v} \leftarrow V_T^{n+\frac{2}{3}}, \mathbf{t}$ at RHS $\leftarrow T_F^{n+\frac{1}{3}}, \mathbf{t}$ at LHS $\leftarrow T_F^{n+\frac{2}{3}}$
$\mathbf{v} = \text{inv}(\frac{1}{2} - \frac{1}{4}C)\mathbf{t}$	// $\mathbf{t} \leftarrow T_F^{n+\frac{2}{3}}, \mathbf{v} \leftarrow V_T^{n+1}$
$\mathbf{t} = \mathbf{v} - \mathbf{t}$	// $\mathbf{v} \leftarrow V_T^{n+1}, \mathbf{t}$ at RHS $\leftarrow T_F^{n+\frac{2}{3}}, \mathbf{t}$ at LHS $\leftarrow T_F^{n+1}$
end	

Figure 4.23: Pseudocode of 3-D FLOD method across iterations.

for $n = 0, 1, 2, \dots$	// $n \leftarrow$ time step in the main iteration
$\mathbf{v} = (1 + \frac{1}{2}A)\mathbf{t}$	// $\mathbf{t} \leftarrow T_{\text{LOD}}^n, \mathbf{v} \leftarrow$ RHS of (4.5.9a)
$\mathbf{t} = \text{inv}(1 - \frac{1}{2}A)\mathbf{v}$	// $\mathbf{t} \leftarrow T_{\text{LOD}}^{n+\frac{1}{3}}, \mathbf{v} \leftarrow$ RHS of (4.5.9a)
$\mathbf{v} = (1 + \frac{1}{2}B)\mathbf{t}$	// $\mathbf{t} \leftarrow T_{\text{LOD}}^{n+\frac{1}{3}}, \mathbf{v} \leftarrow$ RHS of (4.5.9b)
$\mathbf{t} = \text{inv}(1 - \frac{1}{2}B)\mathbf{v}$	// $\mathbf{t} \leftarrow T_{\text{LOD}}^{n+\frac{2}{3}}, \mathbf{v} \leftarrow$ RHS of (4.5.9b)
$\mathbf{v} = (1 + \frac{1}{2}C)\mathbf{t}$	// $\mathbf{t} \leftarrow T_{\text{LOD}}^{n+\frac{2}{3}}, \mathbf{v} \leftarrow$ RHS of (4.5.9c)
$\mathbf{t} = \text{inv}(1 - \frac{1}{2}C)\mathbf{v}$	// $\mathbf{t} \leftarrow T_{\text{LOD}}^{n+1}, \mathbf{v} \leftarrow$ RHS of (4.5.9c)
end	

Figure 4.24: Pseudocode of 3-D LOD method across iterations.

Figures 4.23 and 4.24.

Figure 4.23 presents the pseudocode of 3-D FLOD method across iterations. It can be seen that two variables, i.e. $\mathbf{t}$ and $\mathbf{v}$ are required. That is, we are able to reuse the memory spaces such that $\mathbf{t}$ is multi-purpose and may represent $T_F^n, T_F^{n+\frac{1}{3}}, T_F^{n+\frac{2}{3}}$ and T_F^{n+1} . Likewise, $\mathbf{v}$ is multi-purpose and may represent $V_T^{n+\frac{1}{3}}, V_T^{n+\frac{2}{3}}$ and V_T^{n+1} .

The pseudocode of 3-D LOD method in Figure 4.24 also requires two variables, i.e. $\mathbf{t}$ and $\mathbf{v}$ for computation in the simulation. Therefore, it is evident that both LOD and FLOD methods require the same amount of memory. Due to the simplicity and conciseness of the update equations for the FLOD method, it is more appealing than the LOD method of its implementation for solving the heat transfer equation.

```

First Procedure
// Main grid
for j=0:je
  for k=0:ke
    i=0;
    auxiliary Eq. (4.5.15a) with BC for  $V_T|_{-1,j,k}^{n+\frac{1}{3}}$ 
    Eq. (4.5.15b)
    for i=1:ie-1
      auxiliary Eq. (4.5.15a)
      Eq. (4.5.15b)
    end
    i=ie;
    auxiliary Eq. (4.5.15a) with BC for  $V_T|_{ie+1,j,k}^{n+\frac{1}{3}}$ 
    Eq. (4.5.15b)
  end
end
end

```

Figure 4.25: Pseudocode of 3-D FLOD method for the first procedure of iteration.

4.5.7 Boundary Conditions

Next, we shall investigate the boundary of the computation domain for both 3-D LOD and FLOD methods. Similar to Section 4.4.3, by discretizing (2.2.2) at various boundaries ($i=0$, $i=ie$, $j=0$, $j=je$, $k=0$, $k=ke$), the virtual points at various time steps can be expressed as in (4.4.17).

Figures 4.25 and 4.26 show the pseudocode of 3-D FLOD and LOD methods for the first procedure of iteration respectively. For every procedure, the boundary conditions can be implemented together with the main grids within a single for-loop. Using the FLOD method [c.f. Figure 4.25], only two virtual point treatments, $V_T|_{-1,j,k}^{n+\frac{1}{3}}$ and $V_T|_{ie+1,j,k}^{n+\frac{1}{3}}$ (one on each boundary) are required for the update equation on boundaries. For the LOD method [c.f. Figure 4.26], four virtual point treatments $T_{\text{LOD}}|_{-1,j,k}^{n+\frac{1}{3}}$, $T_{\text{LOD}}|_{-1,j,k}^n$, $T_{\text{LOD}}|_{ie+1,j,k}^{n+\frac{1}{3}}$ and $T_{\text{LOD}}|_{ie+1,j,k}^n$ (two on each boundary)

```

First Procedure
// Main grid
for j=0:je
for k=0:ke
    i=0;
        Eq. (4.5.10a) with BCs for  $T_{\text{LOD}}|_{-1,j,k}^{n+\frac{1}{3}}$  &  $T_{\text{LOD}}|_{-1,j,k}^n$ 
    for i=1:ie-1
        Eq. (4.5.10a)
    end
    i=ie;
        Eq. (4.5.10a) with BCs for  $T_{\text{LOD}}|_{ie+1,j,k}^{n+\frac{1}{3}}$  &  $T_{\text{LOD}}|_{ie+1,j,k}^n$ 
    end
end
end

```

Figure 4.26: Pseudocode of 3-D LOD method for the first procedure of iteration.

are required. Moreover, with a reduction in the number of arithmetic operations for the FLOD method, the implementation of the FLOD method is more desirable as compared to the LOD method.

4.5.8 Stability Analysis, Efficiency and Numerical Results

Stability Analysis

To analyze the stability of 3-D DG-ADI and FLOD methods, we consider a simulation layout of a computation domain with piecewise homogeneous medium shown in Figure 4.27. The computation domain has dimension of $12 \times 12 \times 12$ grids with spatial step $\Delta x = \Delta y = \Delta z = 20$ nm and time step Δt specified in terms of γ in (2.2.6). The materials and associated parameters used in the piecewise homogeneous medium are as follows: silicon (thermal conductivity $\kappa = 131$ W/m·K, density $\rho = 2500$ kg/m³, specific heat $C_p = 700$ J/kg·K), silver (thermal conductivity $\kappa = 315.82$ W/m·K, density $\rho = 8954.5$ kg/m³, specific heat $C_p = 383.6$ J/kg·K) and

Figure 4.27: Layout of a computation domain with piecewise homogeneous medium.

alumina (thermal conductivity $\kappa = 20$ W/m·K, density $\rho = 2699$ kg/m³, specific heat $C_p = 901$ J/kg·K). The computation domain is bounded by the convection boundary conditions in (4.4.17) where the effective heat transfer coefficients for all boundaries (h_x^- , h_x^+ , h_y^- , h_y^+ , h_z^- and h_z^+) are set as 2×10^4 W/m²·K. This domain is surrounded by a heat bath with ambient temperature, T_∞ assumed to be 26.85 °C. The heat generation rate (W/m³) in silver and alumina is given by

$$g_{i,j,k}^n = \begin{cases} \left(1 + \cos(2\pi f_0 t + \pi)\right) \times 0.5 \times 10^{14} & \text{if } 0 \leq t \leq \frac{1}{2f_0} \\ 1.0 \times 10^{14} & \text{otherwise} \end{cases} \quad (4.5.16)$$

where $f_0 = 1.38 \times 10^5$ Hz. Note that the maximum value of the heat generation rate is set at 1.0×10^{14} W/m³ [129].

We now investigate the eigenvalues of the amplification matrix of the system for

Figure 4.28: Scatter plot of eigenvalues of the amplification matrix for 3-D DG-ADI method [c.f. (4.5.8)] for various γ .

DG-ADI and FLOD methods. Note that the von-Neumann Fourier method is not applicable here due to the inhomogeneity of the computation domain. Instead, we need to consider the amplification matrix of the whole computation domain. Figure 4.28 shows the scatter plot of eigenvalues of the amplification matrix for 3-D DG-ADI method [c.f. (4.5.8)] for various γ . By gradually increasing γ , it is found that some eigenvalues start to deviate outside the unit semi circle around $\gamma = 30$. This further substantiates the potential instability for the DG-ADI method. On the other hand, Figure 4.29 shows that all the eigenvalues for 3-D FLOD method [c.f. (4.5.15)] are located inside the unit semi circle. This verifies the stability of our FLOD method.

Figure 4.30 shows the transient temperature at observation point ($i = 4, j = 4, k = 4$) computed using 3-D DG-ADI method [c.f. (4.5.8)] at $\gamma = 30$. As the eigenvalues of the DG-ADI method in Figure 4.28 shows that it is potentially unstable, the

Figure 4.29: Scatter plot of eigenvalues of the amplification matrix for 3-D FLOD method [c.f. (4.5.15)] for various γ .

transient temperature that goes unbounded over time in Figure 4.30 is due to the potentially unstable DG-ADI method and not late time instability. However, there is no instability in the computed transient temperatures for the FLOD method [c.f. (4.5.15)] for $\gamma \geq \frac{1}{2}$, as shown in Figure 4.31. In addition, the explicit method is included for comparison. The temperature of the piecewise homogeneous medium has reached steady state. The results computed by the FLOD method are similar with that of the explicit method, which validates our proposed method.

Efficiency

Next, we conduct numerical experiments for a range of computation domains from $50 \times 50 \times 50$ to $250 \times 250 \times 250$ grids with $\gamma = 50$. The programs have been compiled using Microsoft Visual C++ under Microsoft Windows 7 operating system (OS) running on Intel Dual Core 2.66 GHz processor platform.

Figure 4.30: Transient temperature at observation point $(i = 4, j = 4, k = 4)$ computed using 3-D DG-ADI method [c.f. (4.5.8)], $\gamma = 30$.

Figure 4.31: Transient temperature at observation point $(i = 4, j = 4, k = 4)$ computed using 3-D FLOD method [c.f. (4.5.15)], for various γ .

Table 4.4: Efficiency gains of the FLOD method over LOD, DG-ADI and explicit methods for various computation domains ($\gamma=50$)

Domain Size	Efficiency Gain		
	FLOD	FLOD	FLOD
	vs LOD	vs DG-ADI	vs Explicit
$50 \times 50 \times 50$	1.43	4.10	91.93
$100 \times 100 \times 100$	1.58	3.65	104.14
$150 \times 150 \times 150$	1.35	3.30	93.85
$200 \times 200 \times 200$	1.35	3.29	100.96
$250 \times 250 \times 250$	1.34	3.31	100.60

Table 4.4 shows the CPU efficiency gains of the FLOD method over LOD, DG-ADI and explicit methods for various computation domains. It can be seen that the efficiency gain of the FLOD method over LOD method ranges from 1.3 to 1.6. This is due to the operator-free RHS for the FLOD method, [c.f. (4.5.15)], which reduces the number of arithmetic operations compared to that of the LOD method, [c.f. (4.5.10)]. If the FLOD method is compared to the DG-ADI method, the efficiency gain increases to 3 to 4 times as the DG-ADI method involves even larger amount of RHS operators, [c.f. (4.5.8)]. If the FLOD method is further compared to the explicit method, the efficiency gain improves up to 100 times. This is because the explicit method has its time step size restricted by the minimum cell size in the computation domain, while the FLOD method has its time step size 100 times ($\gamma = 50$) larger.

Numerical Results

For the numerical simulations, we consider a 3-D chip with layout of its hierarchy function blocks and power density shown in Figure 4.32. The chip size is

Figure 4.32: A chip layout with the power density in each hierarchical function block.

$2.4 \text{ mm} \times 2.8 \text{ mm} \times 0.4 \text{ mm}$ and the spatial steps are $\Delta x = \Delta y = \Delta z = 20 \text{ } \mu\text{m}$. The effective heat transfer coefficients h_x^- , h_x^+ , h_y^- and h_y^+ for the convection boundary conditions on the sides of the chip are set as $2.5 \times 10^3 \text{ W/m}^2\cdot\text{K}$. The primary heat transfer path h_z^+ and the secondary heat transfer path h_z^- are set as $3 \times 10^3 \text{ W/m}^2\cdot\text{K}$ and $4 \times 10^4 \text{ W/m}^2\cdot\text{K}$, respectively.

Figure 4.33 shows the temperature profile of the chip at 0.1 s. The highest temperature is about $107 \text{ } ^\circ\text{C}$. By taking an observation point at $i = 60, j = 60, k = 12$ on the substrate, the transient temperature results are shown in Figure 4.34. The temperature of the chip has reached steady state. It can be seen that as γ gets larger, the FLOD curves deviate further away from the explicit method.

Next, we investigate the error of the FLOD method by computing the relative

Figure 4.33: Temperature profile of the chip at 0.1 s.

Figure 4.34: Transient temperature results using 3-D FLOD method at observation point ($i = 60, j = 60, k = 12$) with various γ .

Figure 4.35: Relative maximum error for 3-D FLOD method with various γ .

Table 4.5: CPU time and relative maximum error of the FLOD method

γ	0.5	5	25	50
Explicit	5971.2s	N.A.	N.A.	N.A.
FLOD	6595.2s	643.9s	129.2s	64.3s
Eff. Gain	0.9	9.3	46.2	92.9
Relative Max. Error	9.34×10^{-5}	9.66×10^{-4}	4.76×10^{-3}	1.08×10^{-2}

maximum error defined as

$$\text{Error} = \frac{\left\| T_F|_{i,j,k} - T_{\text{explicit}}|_{i,j,k} \right\|_{\infty}}{\left\| T_{\text{explicit}}|_{i,j,k} \right\|_{\infty}}. \quad (4.5.17)$$

Here, T_F is the temperature recorded using the FLOD method with various γ and T_{explicit} is that recorded using the explicit method with $\gamma = 0.5$.

Figure 4.35 shows the relative maximum error for 3-D FLOD method with various γ . The CPU time and relative maximum error are further shown in Table 4.5. It can

Figure 4.36: Floorplan of Alpha 21364 processor.

been seen that as the γ increases, the relative maximum error increases. At $\gamma = 5$, the FLOD method allows us to choose Δt to be 10 times larger than that of the explicit method, at the expense of 9.66×10^{-4} relative maximum error while having an efficiency gain of 9.3. Even at $\gamma = 50$, the FLOD method allows us to choose Δt to be 100 times larger than that of the explicit method, at the expense of 1.08×10^{-2} relative maximum error while having an efficiency gain of 92.9. This shows that the FLOD method exhibits good trade-off between accuracy and efficiency.

We further simulate another example which floorplan closely resembles the Alpha 21364 processor [143–145] shown in Figure 4.36. The processor chip follows the dimension of $3.3 \text{ mm} \times 3.3 \text{ mm} \times 0.5 \text{ mm}$ with a 20 nm power source layer included on top of the processor chip [125, 129, 130]. The power density in each hierarchical function block of the Alpha 21364 processor is provided in Figure 4.37. Figure 4.38 shows the temperature profile at the center of the chip in steady state. It can be seen that the core function blocks are the region with the higher temperature where the arithmetics operations and instructions are executed.

Figure 4.37: Power density in each hierarchical function block.

Figure 4.38: Temperature profile of the chip in steady state.

4.6 Conclusions

This chapter has presented a stable 2-D FADI method for solving the heat transfer equation. The formulation has taken into consideration both Laplacian and gradient terms of the heat transfer equation. The potential instability of the conventional DG-ADI method caused by the gradient terms within inhomogeneous media is first alleviated. The proposed stabilized DG-ADI method is then cast into the (stabilized) PR-ADI method in compact form, and further formulated into the stable and efficient FADI method with operator-free RHS. This results in simpler and more concise update equations. The temperatures resulted from DG-ADI and PR-ADI methods differ in their intermediate values, while the FADI method has its temperatures being the same as the PR-ADI method. These three methods can be related to one another and their relations have been discussed. Stability analysis by means of analyzing the eigenvalues of the reduced amplification matrix has verified the stability of the FADI method, while the potential instability of the conventional DG-ADI method for inhomogeneous media has been demonstrated. Furthermore, numerical results have justified the high efficiency gains achievable for the FADI method over DG-ADI and PR-ADI methods.

Next, the chapter has presented the GPU-accelerated FADI method for efficient thermal simulation of ICs with microchannel cooling. The convection heat flux due to fluid motion has been included in the heat transfer equation. Subsequently, the FADI method has been introduced for thermal simulation with its RHS free of operators, leading to computationally efficient update equations. To further accelerate the FADI method, the GPU has been utilized through CUDA implementation. It has been shown that high efficiency gain can be achieved through large time step size and data parallelism. Numerical results have further ascertained the cooling effect of the microchannels.

The conventional 3-D DG-ADI method in homogeneous media comes with a high complexity and increase memory variables in its implementation. To overcome these complications, we have presented a 3-D efficient ADI method for thermal modeling. The DG-ADI method has been formulated into the efficient ADI method with single operator and heat generation input on the RHS of the first procedure, reducing the number of arithmetic operations to the minimal. The efficient ADI method has been compared and validated with the DG-ADI method through numerical simulations. Using the efficient ADI method, there has been a decrease in the number of memory variables required, which reduces the memory space and memory indexing overhead. Comparing the CPU computation time of DG-ADI and efficient ADI methods, it is ascertained that high efficiency gain can be achieved for the latter.

To overcome the potential instability of 3-D DG-ADI method within inhomogeneous media, we have presented two stable LOD methods for thermal simulation. The first stable LOD method for solving the heat transfer equation takes into account both Laplacian and gradient terms. On the other hand, the second stable LOD method involves the discretization of the heat transfer equation with gradient terms being absorbed into the finite-difference operator directly without being expanded out. The LOD method is then cast into the compact form and formulated into the FLOD method with operator-free RHS, leading to computationally efficient update equations. Stability analysis by means of analyzing the eigenvalues of the amplification matrix has verified the stability of the FLOD method. Additionally, the potential instability of the conventional DG-ADI method for inhomogeneous media has been demonstrated. Numerical experiments have justified that substantial gain in the overall efficiency has been achieved for the FLOD method over LOD method. Furthermore, the relative maximum error of the FLOD method has been demonstrated, which exhibits good trade-off between accuracy and efficiency. To show the effectiveness of the FLOD method, the heat distribution of the chip of the

closely resembled Alpha 21364 processor chip has been presented and analyzed .

Chapter 5

Extension of Stable and Efficient ADI-FDTD Method for Schrödinger Equation

5.1 Introduction

The FDTD technique can be used in various branches of physics namely, electromagnetics, thermodynamics and quantum mechanics. In Chapter 3, the application of FDTD methods are used in electromagnetics for solving Maxwell's equations which are hyperbolic partial differential equations. This is followed by Chapter 4, the FDTD methods are used in thermodynamics to solve the parabolic partial differential equation of the heat transfer equation. Based on the FDTD technique used for solving Maxwell's equations, it has been extended to solve the heat transfer equation for heat analysis of electrical devices such as IC chips. Another branch of physics which is of great importance is the quantum mechanics. In this chapter, by using the FDTD technique for solving the heat transfer equation, this technique can be extended to solve the Schrödinger equation which is also a parabolic partial differential equation with complex variables.

The time-dependent Schrödinger equation forms the basis of quantum mechanics in

modern physics. It is a partial differential equation which describes the behavior of physical system over time. This equation can be applied to a wide range of applications, such as optics and optoelectronic devices [146–148], hydrodynamics [149, 150], thermodynamics processes [151], etc.

Various methods [152–158] have been proposed to solve the Schrödinger equation. In [152], the meshless local Petrov-Galerkin (MLPG) method based on local weak form and moving least squares (MLS) approximation has been proposed to solve the Schrödinger equation. The meshless local boundary integral equation (LBIE) method developed in [153] for numerical solution of the Schrödinger equation is based on LBIE with MLS approximation. The compact boundary value method introduced in [154] to solve the Schrödinger equation involves a combination of compact finite-difference approximation of fourth-order and fourth-order boundary value method. In [155], the time-space pseudo-spectral method based on Chebyshev-Gauss-Lobatto quadrature points is developed for numerical solution of the Schrödinger equation. The finite element method introduced in [156–158] is used to analyze the Schrödinger equation in different domains. The quantum FDTD method can be used to solve the Schrödinger equation. Based on the quantum FDTD method, various numerical schemes have been proposed, such as the explicit finite-difference method [75–83, 159, 160], CN method [88–90], SS method [93–95] and ADI method [91, 92, 161–163]. The ADI methods in [161] are based on Peaceman and Rachford second-order ADI method and Mitchell and Fairweather fourth-order ADI method for solving 2-D time dependent diffusion equation with non-local boundary conditions. The Noye and Hayman fourth-order ADI method developed in [162] is used to solve 2-D time dependent diffusion equation with an integral condition. In [163], the second-order and fourth-order CN-ADI schemes as well as the linearized ADI scheme are introduced to solve the Schrödinger equation.

While various methods can be used to obtain the solution of the Schrödinger time-dependent equation, precautions have to be taken on the potential function, $\hat{\omega}$ when it is singular. This is because the potential function will render the computation of the Schrödinger equation to become unstable. Using the quantum FDTD method, the computation domain is first discretized into grids. The discretization of the computation domain can be arranged such that the singular potential function is avoided.

For fast and simple implementation, one can consider using the explicit method. However, this comes with a drawback of stability constraint [77, 78, 84, 85] restricted by the minimum spatial step of the computation domain. This limitation can be overcome through the implementation of the Tri-ADI method [91, 92, 161–163], which involves a tridiagonal system of equations. Nevertheless, the conventional Tri-ADI method involves direct computations of the complex wave function, resulting in lower efficiency as the computations of complex variables are rather taxing and time consuming. Furthermore, all the implicit methods so far still contain operators on the RHS, which call for considerable arithmetic operations.

To improve the overall efficiency, we propose a novel Penta-ADI method for the Schrödinger equation. Through the separation of the complex wave function into real and imaginary parts, a pentadiagonal system of equations for the ADI method is obtained, which results in our Penta-ADI method. Using this algorithm, the overall efficiency improves as no complex variables and operations is required. To further simplify the algorithm, the concept of FADI method in Section 3.2.1 is borrowed and extended from the electromagnetics. This algorithm has its RHS free of matrix operators, leading to computationally efficient update equations. Note that for the FADI method in electromagnetics [1, 2], it is not directly applicable here because it involves tridiagonal system of equations. In our case here for the Schrödinger

equation, it involves pentadiagonal system of equations, which shall be called as Penta-FADI method. As the Penta-FADI method involves five stencils on the LHS of pentadiagonal update equations, special treatments that are required for the implementation of the Dirichlet boundary condition will be discussed. Using the Penta-FADI method, a high efficiency gain can be achieved over the conventional Tri-ADI method.

5.2 Pentadiagonal ADI (Penta-ADI) Method

To overcome the deficiency of the complex wave function Ψ in the Tri-ADI method (2.3.14), we separate the complex wave function into real and imaginary parts given in (2.3.11). Upon doing so, we can now convert (2.3.14) into a compact matrix form as

$$\left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{A}_Q\right) \Psi^{n+\frac{1}{2}} = \left(\mathbf{I}_2 + \frac{\Delta t}{2} \mathbf{B}_Q\right) \Psi^n \quad (5.2.1a)$$

$$\left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{B}_Q\right) \Psi^{n+1} = \left(\mathbf{I}_2 + \frac{\Delta t}{2} \mathbf{A}_Q\right) \Psi^{n+\frac{1}{2}} \quad (5.2.1b)$$

where

$$\Psi = \begin{bmatrix} \Psi_R \\ \Psi_I \end{bmatrix}, \quad \mathbf{A}_Q = \begin{bmatrix} 0 & -\hat{H}_x \\ \hat{H}_x & 0 \end{bmatrix}, \quad \mathbf{B}_Q = \begin{bmatrix} 0 & -\hat{H}_y \\ \hat{H}_y & 0 \end{bmatrix}, \quad \hat{H}_{\tilde{\xi}} = \frac{\hbar}{2m_e} \frac{\partial^2}{\partial \tilde{\xi}^2} - \frac{\hat{\omega}_{i,j}}{2\hbar}$$

and $\mathbf{I}_2$ is an 2×2 identity matrix.

By discretizing (5.2.1a), we get

$$\begin{aligned} \Psi_R|_{i,j}^{n+\frac{1}{2}} + \hat{a}_{2x} \left(\Psi_I|_{i+1,j}^{n+\frac{1}{2}} + \Psi_I|_{i-1,j}^{n+\frac{1}{2}} \right) - (\hat{a}_{1x} + \hat{b}_{i,j}) \Psi_I|_{i,j}^{n+\frac{1}{2}} \\ = \Psi_R|_{i,j}^n - \hat{a}_{2y} \left(\Psi_I|_{i,j+1}^n + \Psi_I|_{i,j-1}^n \right) + (\hat{a}_{1y} + \hat{b}_{i,j}) \Psi_I|_{i,j}^n \end{aligned} \quad (5.2.2a)$$

$$\Psi_I|_{i,j}^{n+\frac{1}{2}} - \hat{a}_{2x} \left(\Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \right) + (\hat{a}_{1x} + \hat{b}_{i,j}) \Psi_R|_{i,j}^{n+\frac{1}{2}}$$

$$= \Psi_I|_{i,j}^n + \widehat{a}_{2y} \left(\Psi_R|_{i,j+1}^n + \Psi_R|_{i,j-1}^n \right) - (\widehat{a}_{1y} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^n. \quad (5.2.2b)$$

By substituting (5.2.2b) into (5.2.2a), we can obtain the implicit pentadiagonal update equation for the first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned} & \left[1 + 2\widehat{a}_{2x}^2 + (\widehat{a}_{1x} + \widehat{b}_{i,j})^2 \right] \Psi_R|_{i,j}^{n+\frac{1}{2}} - \widehat{a}_{2x} \left[2\widehat{a}_{1x} + \widehat{b}_{i-1,j} + \widehat{b}_{i,j} \right] \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \\ & - \widehat{a}_{2x} \left[2\widehat{a}_{1x} + \widehat{b}_{i,j} + \widehat{b}_{i+1,j} \right] \Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \widehat{a}_{2x}^2 \left(\Psi_R|_{i+2,j}^{n+\frac{1}{2}} + \Psi_R|_{i-2,j}^{n+\frac{1}{2}} \right) \\ & = \left[1 - (\widehat{a}_{1x} + \widehat{b}_{i,j})(\widehat{a}_{1y} + \widehat{b}_{i,j}) \right] \Psi_R|_{i,j}^n + \widehat{a}_{2x}(\widehat{a}_{1y} + \widehat{b}_{i+1,j}) \Psi_R|_{i+1,j}^n \\ & + \widehat{a}_{2x}(\widehat{a}_{1y} + \widehat{b}_{i-1,j}) \Psi_R|_{i-1,j}^n + \widehat{a}_{2y}(\widehat{a}_{1x} + \widehat{b}_{i,j}) \left(\Psi_R|_{i,j+1}^n + \Psi_R|_{i,j-1}^n \right) \\ & - \widehat{a}_{2x}\widehat{a}_{2y} \left(\Psi_R|_{i+1,j+1}^n + \Psi_R|_{i-1,j+1}^n + \Psi_R|_{i-1,j-1}^n + \Psi_R|_{i+1,j-1}^n \right) \\ & + (\widehat{a}_{1x} + \widehat{a}_{1y} + 2\widehat{b}_{i,j}) \Psi_I|_{i,j}^n - \widehat{a}_{2x} \left(\Psi_I|_{i+1,j}^n + \Psi_I|_{i-1,j}^n \right) - \widehat{a}_{2y} \left(\Psi_I|_{i,j+1}^n + \Psi_I|_{i,j-1}^n \right). \end{aligned} \quad (5.2.3)$$

$\Psi_I^{n+\frac{1}{2}}$ can then be updated explicitly through (5.2.2b) as

$$\begin{aligned} \Psi_I|_{i,j}^{n+\frac{1}{2}} &= \Psi_I|_{i,j}^n + \widehat{a}_{2y} \left(\Psi_R|_{i,j+1}^n + \Psi_R|_{i,j-1}^n \right) - (\widehat{a}_{1y} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^n \\ &+ \widehat{a}_{2x} \left(\Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \right) - (\widehat{a}_{1x} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^{n+\frac{1}{2}}. \end{aligned} \quad (5.2.4)$$

Similarly, we can obtain the implicit pentadiagonal update equation for the second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} & \left[1 + 2\widehat{a}_{2y}^2 + (\widehat{a}_{1y} + \widehat{b}_{i,j})^2 \right] \Psi_R|_{i,j}^{n+1} - \widehat{a}_{2y} \left[2\widehat{a}_{1y} + \widehat{b}_{i,j-1} + \widehat{b}_{i,j} \right] \Psi_R|_{i,j-1}^{n+1} \\ & - \widehat{a}_{2y} \left[2\widehat{a}_{1y} + \widehat{b}_{i,j} + \widehat{b}_{i,j+1} \right] \Psi_R|_{i,j+1}^{n+1} + \widehat{a}_{2y}^2 \left(\Psi_R|_{i,j+2}^{n+1} + \Psi_R|_{i,j-2}^{n+1} \right) \\ & = \left[1 - (\widehat{a}_{1y} + \widehat{b}_{i,j})(\widehat{a}_{1x} + \widehat{b}_{i,j}) \right] \Psi_R|_{i,j}^{n+\frac{1}{2}} + \widehat{a}_{2y}(\widehat{a}_{1x} + \widehat{b}_{i,j+1}) \Psi_R|_{i,j+1}^{n+\frac{1}{2}} \\ & + \widehat{a}_{2y}(\widehat{a}_{1x} + \widehat{b}_{i,j-1}) \Psi_R|_{i,j-1}^{n+\frac{1}{2}} + \widehat{a}_{2x}(\widehat{a}_{1y} + \widehat{b}_{i,j}) \left(\Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \right) \\ & - \widehat{a}_{2x}\widehat{a}_{2y} \left(\Psi_R|_{i+1,j+1}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j+1}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j-1}^{n+\frac{1}{2}} + \Psi_R|_{i+1,j-1}^{n+\frac{1}{2}} \right) \\ & + (\widehat{a}_{1x} + \widehat{a}_{1y} + 2\widehat{b}_{i,j}) \Psi_I|_{i,j}^{n+\frac{1}{2}} - \widehat{a}_{2y} \left(\Psi_I|_{i,j+1}^{n+\frac{1}{2}} + \Psi_I|_{i,j-1}^{n+\frac{1}{2}} \right) - \widehat{a}_{2x} \left(\Psi_I|_{i+1,j}^{n+\frac{1}{2}} + \Psi_I|_{i-1,j}^{n+\frac{1}{2}} \right). \end{aligned} \quad (5.2.5)$$

Ψ_I^{n+1} can then be updated explicitly through

$$\begin{aligned}\Psi_I|_{i,j}^{n+1} = & \Psi_I|_{i,j}^{n+\frac{1}{2}} + \widehat{a}_{2x} \left(\Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \right) - (\widehat{a}_{1x} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^{n+\frac{1}{2}} \\ & + \widehat{a}_{2y} \left(\Psi_R|_{i,j+1}^{n+1} + \Psi_R|_{i,j-1}^{n+1} \right) - (\widehat{a}_{1y} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^{n+1}.\end{aligned}\quad (5.2.6)$$

By applying separation of the complex wave function in (2.3.11), we obtain a novel pentadiagonal system of equations on the LHS of (5.2.3) and (5.2.5), which result in our Penta-ADI method. Note that for 2-D system, the order of computational complexity for solving pentadiagonal system of equations is $O(n^2)$. As this algorithm does not involve complex variables, the computation is much simpler and less time consuming. The efficiency gain will be further demonstrated in Section 5.5.

5.3 Pentadiagonal FADI (Penta-FADI) Method

To further simplify the complexity of the RHS update equations in (5.2.1), we now introduce an auxiliary matrix operator $\tilde{\mathbf{V}}_Q$ and rewrite (5.2.1) as

$$\tilde{\mathbf{V}}_Q^n = \left(\mathbf{I}_2 + \frac{\Delta t}{2} \mathbf{B}_Q \right) \Psi^n \quad (5.3.1a)$$

$$\left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{A}_Q \right) \Psi^{n+\frac{1}{2}} = \tilde{\mathbf{V}}_Q^n \quad (5.3.1b)$$

$$\tilde{\mathbf{V}}_Q^{n+\frac{1}{2}} = \left(\mathbf{I}_2 + \frac{\Delta t}{2} \mathbf{A}_Q \right) \Psi^{n+\frac{1}{2}} \quad (5.3.1c)$$

$$\left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{B}_Q \right) \Psi^{n+1} = \tilde{\mathbf{V}}_Q^{n+\frac{1}{2}} \quad (5.3.1d)$$

where

$$\tilde{\mathbf{V}}_Q = \begin{bmatrix} \tilde{V}_R & \tilde{V}_I \end{bmatrix}^T.$$

First, we alter (5.3.1d) by one time step backward to get

$$\tilde{\mathbf{V}}_Q^{n-\frac{1}{2}} = \left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{B}_Q \right) \Psi^n. \quad (5.3.2)$$

Further manipulation of (5.3.1a), we can obtain

$$\begin{aligned}
 \tilde{\mathbf{V}}_Q^n &= \left(\mathbf{I}_2 + \frac{\Delta t}{2} \mathbf{B}_Q \right) \Psi^n \\
 &= 2\Psi^n - \left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{B}_Q \right) \Psi^n \\
 &= 2\Psi^n - \tilde{\mathbf{V}}_Q^{n-\frac{1}{2}}.
 \end{aligned} \tag{5.3.3}$$

Similarly, through (5.3.1b), we are able to reduce (5.3.1c) into

$$\begin{aligned}
 \tilde{\mathbf{V}}_Q^{n+\frac{1}{2}} &= \left(\mathbf{I}_2 + \frac{\Delta t}{2} \mathbf{A}_Q \right) \Psi^{n+\frac{1}{2}} \\
 &= 2\Psi^{n+\frac{1}{2}} - \left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{A}_Q \right) \Psi^{n+\frac{1}{2}} \\
 &= 2\Psi^{n+\frac{1}{2}} - \tilde{\mathbf{V}}_Q^{n+\frac{1}{2}}.
 \end{aligned} \tag{5.3.4}$$

Combining (5.3.3) and (5.3.4), we have simplified the algorithm in (5.3.1) to

$$\tilde{\mathbf{V}}_Q^n = 2\Psi^n - \tilde{\mathbf{V}}_Q^{n-\frac{1}{2}} \tag{5.3.5a}$$

$$\left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{A}_Q \right) \Psi^{n+\frac{1}{2}} = \tilde{\mathbf{V}}_Q^n \tag{5.3.5b}$$

$$\tilde{\mathbf{V}}_Q^{n+\frac{1}{2}} = 2\Psi^{n+\frac{1}{2}} - \tilde{\mathbf{V}}_Q^n \tag{5.3.5c}$$

$$\left(\mathbf{I}_2 - \frac{\Delta t}{2} \mathbf{B}_Q \right) \Psi^{n+1} = \tilde{\mathbf{V}}_Q^{n+\frac{1}{2}}. \tag{5.3.5d}$$

By re-definition of field variables

$$\mathbf{V}_Q^{n-\frac{1}{2}} = \frac{1}{2} \tilde{\mathbf{V}}_Q^{n-\frac{1}{2}}, \quad \mathbf{V}_Q^n = \frac{1}{2} \tilde{\mathbf{V}}_Q^n, \quad \mathbf{V}_Q^{n+\frac{1}{2}} = \frac{1}{2} \tilde{\mathbf{V}}_Q^{n+\frac{1}{2}},$$

we obtain the final update procedures as

$$\mathbf{V}_Q^n = \Psi^n - \mathbf{V}_Q^{n-\frac{1}{2}} \tag{5.3.6a}$$

$$\left(\frac{1}{2} \mathbf{I}_2 - \frac{\Delta t}{4} \mathbf{A}_Q \right) \Psi^{n+\frac{1}{2}} = \mathbf{V}_Q^n \tag{5.3.6b}$$

$$\mathbf{V}_Q^{n+\frac{1}{2}} = \Psi^{n+\frac{1}{2}} - \mathbf{V}_Q^n \tag{5.3.6c}$$

$$\left(\frac{1}{2} \mathbf{I}_2 - \frac{\Delta t}{4} \mathbf{B}_Q \right) \Psi^{n+1} = \mathbf{V}_Q^{n+\frac{1}{2}} \tag{5.3.6d}$$

with initialization

$$\mathbf{V}_Q^{-\frac{1}{2}} = \left(\frac{1}{2} \mathbf{I}_2 - \frac{\Delta t}{4} \mathbf{B}_Q \right) \Psi^0. \quad (5.3.7)$$

where the auxiliary variables

$$\mathbf{V}_Q = \begin{bmatrix} V_R & V_I \end{bmatrix}^T.$$

The algorithm (5.3.6) has now been reduced with no matrix operators $\mathbf{A}_Q$ or $\mathbf{B}_Q$ on the RHS, leading to computationally efficient update equations. Such concept of fundamental scheme is borrowed and extended from the electromagnetics in Section 3.2.1. Note that for the FADI method for electromagnetics in (3.2.10), it is not directly applicable here because it involves tridiagonal system of equations. In our case here for the Schrödinger equation, it involves pentadiagonal system of equations, which shall be called as Penta-FADI method.

To obtain the implicit pentadiagonal update equations, we apply central approximation to (5.3.6b) and upon some manipulations, we have

$$\frac{1}{2} \Psi_R|_{i,j}^{n+\frac{1}{2}} + \hat{a}_{4x} \left(\Psi_I|_{i+1,j}^{n+\frac{1}{2}} + \Psi_I|_{i-1,j}^{n+\frac{1}{2}} \right) - \frac{\hat{a}_{1x} + \hat{b}_{i,j}}{2} \Psi_I|_{i,j}^{n+\frac{1}{2}} = V_R|_{i,j}^n \quad (5.3.8a)$$

$$\frac{1}{2} \Psi_I|_{i,j}^{n+\frac{1}{2}} - \hat{a}_{4x} \left(\Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \right) + \frac{\hat{a}_{1x} + \hat{b}_{i,j}}{2} \Psi_R|_{i,j}^{n+\frac{1}{2}} = V_I|_{i,j}^n. \quad (5.3.8b)$$

By substituting (5.3.8b) into (5.3.8a), we can obtain the update equations of (5.3.6a)-(5.3.6b) for the first procedure from n to $n + \frac{1}{2}$:

$$V_R|_{i,j}^n = \Psi_R|_{i,j}^n - V_R|_{i,j}^{n-\frac{1}{2}} \quad (5.3.9)$$

$$V_I|_{i,j}^n = \Psi_I|_{i,j}^n - V_I|_{i,j}^{n-\frac{1}{2}} \quad (5.3.10)$$

$$\begin{aligned} & \frac{1}{2} \left[1 + 2\hat{a}_{2x}^2 + (\hat{a}_{1x} + \hat{b}_{i,j})^2 \right] \Psi_R|_{i,j}^{n+\frac{1}{2}} - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{i-1,j} + \hat{b}_{i,j} \right] \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \\ & - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{i,j} + \hat{b}_{i+1,j} \right] \Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \frac{\hat{a}_{2x}^2}{2} \left(\Psi_R|_{i+2,j}^{n+\frac{1}{2}} + \Psi_R|_{i-2,j}^{n+\frac{1}{2}} \right) \end{aligned}$$

$$= V_R|_{i,j}^n - \widehat{a}_{2x} \left(V_I|_{i+1,j}^n + V_I|_{i-1,j}^n \right) + (\widehat{a}_{1x} + \widehat{b}_{i,j}) V_I|_{i,j}^n. \quad (5.3.11)$$

$\Psi_I^{n+\frac{1}{2}}$ can then be updated explicitly through (5.3.8b) as

$$\Psi_I|_{i,j}^{n+\frac{1}{2}} = 2V_I|_{i,j}^n + \widehat{a}_{2x} \left(\Psi_R|_{i+1,j}^{n+\frac{1}{2}} + \Psi_R|_{i-1,j}^{n+\frac{1}{2}} \right) - (\widehat{a}_{1x} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^{n+\frac{1}{2}}. \quad (5.3.12)$$

Similarly, we can obtain the update equations of (5.3.6c)-(5.3.6d) for the second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$V_R|_{i,j}^{n+\frac{1}{2}} = \Psi_R|_{i,j}^{n+\frac{1}{2}} - V_R|_{i,j}^n \quad (5.3.13)$$

$$V_I|_{i,j}^{n+\frac{1}{2}} = \Psi_I|_{i,j}^{n+\frac{1}{2}} - V_I|_{i,j}^n \quad (5.3.14)$$

$$\begin{aligned} & \frac{1}{2} \left[1 + 2\widehat{a}_{2y}^2 + (\widehat{a}_{1y} + \widehat{b}_{i,j})^2 \right] \Psi_R|_{i,j}^{n+1} - \widehat{a}_{4y} \left[2\widehat{a}_{1y} + \widehat{b}_{i,j-1} + \widehat{b}_{i,j} \right] \Psi_R|_{i,j-1}^{n+1} \\ & - \widehat{a}_{4y} \left[2\widehat{a}_{1y} + \widehat{b}_{i,j} + \widehat{b}_{i,j+1} \right] \Psi_R|_{i,j+1}^{n+1} + \frac{\widehat{a}_{2y}^2}{2} \left(\Psi_R|_{i,j+2}^{n+1} + \Psi_R|_{i,j-2}^{n+1} \right) \\ & = V_R|_{i,j}^{n+\frac{1}{2}} - \widehat{a}_{2y} \left(V_I|_{i,j+1}^{n+\frac{1}{2}} + V_I|_{i,j-1}^{n+\frac{1}{2}} \right) + (\widehat{a}_{1y} + \widehat{b}_{i,j}) V_I|_{i,j}^{n+\frac{1}{2}}. \end{aligned} \quad (5.3.15)$$

Ψ_I^{n+1} can then be updated explicitly through

$$\Psi_I|_{i,j}^{n+1} = 2V_I|_{i,j}^{n+\frac{1}{2}} + \widehat{a}_{2y} \left(\Psi_R|_{i,j+1}^{n+1} + \Psi_R|_{i,j-1}^{n+1} \right) - (\widehat{a}_{1y} + \widehat{b}_{i,j}) \Psi_R|_{i,j}^{n+1}. \quad (5.3.16)$$

For non-zero initial $\Psi_R|_{i,j}^0$ and $\Psi_I|_{i,j}^0$, we need to apply the initialization as follows:

$$V_R|_{i,j}^{-\frac{1}{2}} = \frac{1}{2} \Psi_R|_{i,j}^0 + \widehat{a}_{4y} \left(\Psi_I|_{i,j+1}^0 + \Psi_I|_{i,j-1}^0 \right) - \frac{\widehat{a}_{1y} + \widehat{b}_{i,j}}{2} \Psi_I|_{i,j}^0 \quad (5.3.17)$$

$$V_I|_{i,j}^{-\frac{1}{2}} = \frac{1}{2} \Psi_I|_{i,j}^0 - \widehat{a}_{4y} \left(\Psi_R|_{i,j+1}^0 + \Psi_R|_{i,j-1}^0 \right) + \frac{\widehat{a}_{1y} + \widehat{b}_{i,j}}{2} \Psi_R|_{i,j}^0. \quad (5.3.18)$$

Comparing (5.2.2)-(5.2.6) with (5.3.8)-(5.3.16), it can be seen that the update equations for the latter have decreased significantly in the number of arithmetic operations and memory indexing. This is due to the matrix-operator-free RHS for the FADI algorithm in (5.3.6).

Using this technique, we are also able to reduce the conventional Tri-ADI method in (2.3.14) into the fundamental form. We rewrite (2.3.14) into the Tri-FADI method as

$$V_Q^n = \Psi^n - V_Q^{n-\frac{1}{2}} \quad (5.3.19a)$$

$$\left(\frac{1}{2} - \frac{\Delta t}{4} A_Q\right) \Psi^{n+\frac{1}{2}} = V_Q^n \quad (5.3.19b)$$

$$V_Q^{n+\frac{1}{2}} = \Psi^{n+\frac{1}{2}} - V_Q^n \quad (5.3.19c)$$

$$\left(\frac{1}{2} - \frac{\Delta t}{4} B_Q\right) \Psi^{n+1} = V_Q^{n+\frac{1}{2}} \quad (5.3.19d)$$

with initialization

$$V_Q^{-\frac{1}{2}} = \left(\frac{1}{2} - \frac{\Delta t}{4} B_Q\right) \Psi^0 \quad (5.3.20)$$

where the auxiliary variable

$$V_Q = V_R + \mathbf{j}V_I \quad (5.3.21)$$

It is worth noting that the Tri-FADI method for the Schrödinger equation in (5.3.19) has its RHS free of complex operators A_Q and B_Q .

By discretizing (5.3.19) using central approximation, we can obtain

$$V_Q|_{i,j}^n = \Psi|_{i,j}^n - V_Q|_{i,j}^{n-\frac{1}{2}} \quad (5.3.22a)$$

$$-\mathbf{j}\hat{a}_{4x} \left(\Psi|_{i-1,j}^{n+\frac{1}{2}} + \Psi|_{i+1,j}^{n+\frac{1}{2}} \right) + \frac{1}{2} \left(1 + \mathbf{j}\hat{a}_{1x} + \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^{n+\frac{1}{2}} = V_Q|_{i,j}^n \quad (5.3.22b)$$

$$V_Q|_{i,j}^{n+\frac{1}{2}} = \Psi|_{i,j}^{n+\frac{1}{2}} - V_Q|_{i,j}^n \quad (5.3.22c)$$

$$-\mathbf{j}\hat{a}_{4y} \left(\Psi|_{i,j-1}^{n+1} + \Psi|_{i,j+1}^{n+1} \right) + \frac{1}{2} \left(1 + \mathbf{j}\hat{a}_{1y} + \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^{n+1} = V_Q|_{i,j}^{n+\frac{1}{2}} \quad (5.3.22d)$$

For non-zero initial Ψ , we apply the initialization as

$$V_Q|_{i,j}^{-\frac{1}{2}} = -\mathbf{j}\hat{a}_{4y} \left(\Psi|_{i,j-1}^0 + \Psi|_{i,j+1}^0 \right) + \frac{1}{2} \left(1 + \mathbf{j}\hat{a}_{1y} + \mathbf{j}\hat{b}_{i,j} \right) \Psi|_{i,j}^0 \quad (5.3.23)$$

Similar to the Penta-FADI method, it can be seen that the update equations for

Figure 5.1: Computation domain of 11×11 grids.

(5.3.22) have a decrease in the number of arithmetic operations and memory indexing as compared to (2.3.15). However, the Tri-FADI method still consists of the complex wave function Ψ which involves real and imaginary parts. Therefore, this algorithm is computationally expensive (refer to Section 5.5.3).

5.4 Boundary Conditions

We have previously mentioned that separating the complex wave function Ψ into real and imaginary parts in (2.3.11) will lead to a pentadiagonal system of equations. Problems arise at the boundaries where the wave function Ψ is located out of the simulated domain. For instance, we consider a computation domain of 11×11 grids shown in Figure 5.1. The grid points ($\bullet$) inside the computation domain are updated using the pentadiagonal systems of equations while the grid points ($\circ$) on the boundaries are updated explicitly by the implementation of the Dirichlet

boundary condition. We will now look at the update equations for the circled grid points in the first procedure from n to $n + \frac{1}{2}$ by forming a pentadiagonal matrix system depicted as

$$\begin{bmatrix}
 \alpha_1 & \gamma_1 & \delta_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 \beta_2 & \alpha_2 & \gamma_2 & \delta_2 & 0 & 0 & 0 & 0 & 0 \\
 \epsilon_3 & \beta_3 & \alpha_3 & \gamma_3 & \delta_3 & 0 & 0 & 0 & 0 \\
 0 & \epsilon_4 & \beta_4 & \alpha_4 & \gamma_4 & \delta_4 & 0 & 0 & 0 \\
 0 & 0 & \epsilon_5 & \beta_5 & \alpha_5 & \gamma_5 & \delta_5 & 0 & 0 \\
 0 & 0 & 0 & \epsilon_6 & \beta_6 & \alpha_6 & \gamma_6 & \delta_6 & 0 \\
 0 & 0 & 0 & 0 & \epsilon_7 & \beta_7 & \alpha_7 & \gamma_7 & \delta_7 \\
 0 & 0 & 0 & 0 & 0 & \epsilon_8 & \beta_8 & \alpha_8 & \gamma_8 \\
 0 & 0 & 0 & 0 & 0 & 0 & \epsilon_9 & \beta_9 & \alpha_9
 \end{bmatrix}
 \begin{bmatrix}
 \Psi_{R|1,1}^{n+\frac{1}{2}} \\
 \Psi_{R|2,1}^{n+\frac{1}{2}} \\
 \Psi_{R|3,1}^{n+\frac{1}{2}} \\
 \Psi_{R|4,1}^{n+\frac{1}{2}} \\
 \Psi_{R|5,1}^{n+\frac{1}{2}} \\
 \Psi_{R|6,1}^{n+\frac{1}{2}} \\
 \Psi_{R|7,1}^{n+\frac{1}{2}} \\
 \Psi_{R|8,1}^{n+\frac{1}{2}} \\
 \Psi_{R|9,1}^{n+\frac{1}{2}}
 \end{bmatrix}
 =
 \begin{bmatrix}
 \Gamma_1 + \widehat{a}_{4x}(\widehat{a}_{1x} + \widehat{b}_{1,1})\Psi_{R|0,1}^{n+\frac{1}{2}} - \widehat{a}_{4x}\Psi_{I|0,1}^{n+\frac{1}{2}} \\
 \Gamma_2 - \epsilon_2\Psi_{R|0,1}^{n+\frac{1}{2}} \\
 \Gamma_3 \\
 \Gamma_4 \\
 \Gamma_5 \\
 \Gamma_6 \\
 \Gamma_7 \\
 \Gamma_8 - \delta_8\Psi_{R|10,1}^{n+\frac{1}{2}} \\
 \Gamma_9 + \widehat{a}_{4x}(\widehat{a}_{1x} + \widehat{b}_{10,1})\Psi_{R|10,1}^{n+\frac{1}{2}} - \widehat{a}_{4x}\Psi_{I|10,1}^{n+\frac{1}{2}}
 \end{bmatrix} \quad (5.4.1)$$

where ϵ , β , α , γ and δ are coefficients of the pentadiagonal matrix while Γ consists of the RHS terms in the update equations.

Note that (5.3.11) is only applicable from $\Psi_{R|3,1}^{n+\frac{1}{2}}$ to $\Psi_{R|7,1}^{n+\frac{1}{2}}$. Special boundary treatments are required for the rest of the update equations. The update equation for $\Psi_{R|1,1}^{n+\frac{1}{2}}$ is obtained by substituting (5.3.8b) into $\Psi_{I|1,1}^{n+\frac{1}{2}}$ and $\Psi_{I|2,1}^{n+\frac{1}{2}}$ of (5.3.8a). $\Psi_{I|0,1}^{n+\frac{1}{2}}$ and $\Psi_{R|0,1}^{n+\frac{1}{2}}$ are then transferred to the RHS of the update equation seen in (5.4.1). The second update equation for $\Psi_{R|2,1}^{n+\frac{1}{2}}$ is the same as (5.3.11), but

requires $\Psi_R|_{0,1}^{n+\frac{1}{2}}$ to be transferred over to the RHS of the update equation. The update equations for $\Psi_R|_{1,1}^{n+\frac{1}{2}}$ and $\Psi_R|_{2,1}^{n+\frac{1}{2}}$ are presented as follows:

$$\begin{aligned} & \frac{1}{2} \left[1 + \hat{a}_{2x}^2 + (\hat{a}_{1x} + \hat{b}_{1,1})^2 \right] \Psi_R|_{1,1}^{n+\frac{1}{2}} - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{1,1} + \hat{b}_{2,1} \right] \Psi_R|_{2,1}^{n+\frac{1}{2}} + \frac{\hat{a}_{2x}^2}{2} \Psi_R|_{3,1}^{n+\frac{1}{2}} \\ & = V_R|_{1,1}^n - \hat{a}_{2x} V_I|_{2,1}^n + (\hat{a}_{1x} + \hat{b}_{1,1}) V_I|_{1,1}^n + \hat{a}_{4x} (\hat{a}_{1x} + \hat{b}_{1,1}) \Psi_R|_{0,1}^{n+\frac{1}{2}} - \hat{a}_{4x} \Psi_I|_{0,1}^{n+\frac{1}{2}} \end{aligned} \quad (5.4.2a)$$

$$\begin{aligned} & \frac{1}{2} \left[1 + 2\hat{a}_{2x}^2 + (\hat{a}_{1x} + \hat{b}_{2,1})^2 \right] \Psi_R|_{2,1}^{n+\frac{1}{2}} - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{1,1} + \hat{b}_{2,1} \right] \Psi_R|_{1,1}^{n+\frac{1}{2}} \\ & - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{2,1} + \hat{b}_{3,1} \right] \Psi_R|_{3,1}^{n+\frac{1}{2}} + \frac{\hat{a}_{2x}^2}{2} \Psi_R|_{4,1}^{n+\frac{1}{2}} \\ & = V_R|_{2,1}^n - \hat{a}_{2x} \left(V_I|_{3,1}^n + V_I|_{1,1}^n \right) + (\hat{a}_{1x} + \hat{b}_{2,1}) V_I|_{2,1}^n - \frac{\hat{a}_{2x}^2}{2} \Psi_R|_{0,1}^{n+\frac{1}{2}}. \end{aligned} \quad (5.4.2b)$$

Similar procedures can be repeated in $\Psi_R|_{8,1}^{n+\frac{1}{2}}$ and $\Psi_R|_{9,1}^{n+\frac{1}{2}}$ to obtain their respective update equations. The update equations are presented as follows:

$$\begin{aligned} & \frac{1}{2} \left[1 + 2\hat{a}_{2x}^2 + (\hat{a}_{1x} + \hat{b}_{8,1})^2 \right] \Psi_R|_{8,1}^{n+\frac{1}{2}} - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{7,1} + \hat{b}_{8,1} \right] \Psi_R|_{7,1}^{n+\frac{1}{2}} \\ & - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{8,1} + \hat{b}_{9,1} \right] \Psi_R|_{9,1}^{n+\frac{1}{2}} + \frac{\hat{a}_{2x}^2}{2} \Psi_R|_{6,1}^{n+\frac{1}{2}} \\ & = V_R|_{8,1}^n - \hat{a}_{2x} \left(V_I|_{9,1}^n + V_I|_{7,1}^n \right) + (\hat{a}_{1x} + \hat{b}_{8,1}) V_I|_{8,1}^n - \frac{\hat{a}_{2x}^2}{2} \Psi_R|_{10,1}^{n+\frac{1}{2}} \end{aligned} \quad (5.4.3a)$$

$$\begin{aligned} & \frac{1}{2} \left[1 + \hat{a}_{2x}^2 + (\hat{a}_{1x} + \hat{b}_{9,1})^2 \right] \Psi_R|_{9,1}^{n+\frac{1}{2}} - \hat{a}_{4x} \left[2\hat{a}_{1x} + \hat{b}_{8,1} + \hat{b}_{9,1} \right] \Psi_R|_{8,1}^{n+\frac{1}{2}} + \frac{\hat{a}_{2x}^2}{2} \Psi_R|_{7,1}^{n+\frac{1}{2}} \\ & = V_R|_{9,1}^n - \hat{a}_{2x} V_I|_{8,1}^n + (\hat{a}_{1x} + \hat{b}_{9,1}) V_I|_{9,1}^n + \hat{a}_{4x} (\hat{a}_{1x} + \hat{b}_{9,1}) \Psi_R|_{10,1}^{n+\frac{1}{2}} - \hat{a}_{4x} \Psi_I|_{10,1}^{n+\frac{1}{2}}. \end{aligned} \quad (5.4.3b)$$

For clarity, the update equations for $\Psi_R|_{1,1}^{n+1}$, $\Psi_R|_{1,2}^{n+1}$, $\Psi_R|_{1,8}^{n+1}$ and $\Psi_R|_{1,9}^{n+1}$ in the second procedure from $n + \frac{1}{2}$ to $n + 1$ are as follows:

$$\begin{aligned} & \frac{1}{2} \left[1 + \hat{a}_{2y}^2 + (\hat{a}_{1y} + \hat{b}_{1,1})^2 \right] \Psi_R|_{1,1}^{n+1} - \hat{a}_{4y} \left[2\hat{a}_{1y} + \hat{b}_{1,1} + \hat{b}_{1,2} \right] \Psi_R|_{1,2}^{n+1} + \frac{\hat{a}_{2y}^2}{2} \Psi_R|_{1,3}^{n+1} \\ & = V_R|_{1,1}^{n+\frac{1}{2}} - \hat{a}_{2y} V_I|_{1,2}^{n+\frac{1}{2}} + (\hat{a}_{1y} + \hat{b}_{1,1}) V_I|_{1,1}^{n+\frac{1}{2}} + \hat{a}_{4y} (\hat{a}_{1y} + \hat{b}_{1,1}) \Psi_R|_{1,0}^{n+1} - \hat{a}_{4y} \Psi_I|_{1,0}^{n+1} \end{aligned} \quad (5.4.4a)$$

$$\frac{1}{2} \left[1 + 2\hat{a}_{2y}^2 + (\hat{a}_{1y} + \hat{b}_{1,2})^2 \right] \Psi_R|_{1,2}^{n+1} - \hat{a}_{4y} \left[2\hat{a}_{1y} + \hat{b}_{1,1} + \hat{b}_{1,2} \right] \Psi_R|_{1,1}^{n+1}$$

$$\begin{aligned}
& -\widehat{a}_{4y} \left[2\widehat{a}_{1y} + \widehat{b}_{1,2} + \widehat{b}_{1,3} \right] \Psi_R|_{1,3}^{n+1} + \frac{\widehat{a}_{2y}^2}{2} \Psi_R|_{1,4}^{n+1} \\
& = V_R|_{1,2}^{n+\frac{1}{2}} - \widehat{a}_{2y} \left(V_I|_{1,3}^{n+\frac{1}{2}} + V_I|_{1,1}^{n+\frac{1}{2}} \right) + (\widehat{a}_{1y} + \widehat{b}_{1,2}) V_I|_{1,2}^{n+\frac{1}{2}} - \frac{\widehat{a}_{2y}^2}{2} \Psi_R|_{1,0}^{n+1}
\end{aligned} \tag{5.4.4b}$$

$$\begin{aligned}
& \frac{1}{2} \left[1 + 2\widehat{a}_{2y}^2 + (\widehat{a}_{1y} + \widehat{b}_{1,8})^2 \right] \Psi_R|_{1,8}^{n+1} - \widehat{a}_{4y} \left[2\widehat{a}_{1y} + \widehat{b}_{1,7} + \widehat{b}_{1,8} \right] \Psi_R|_{1,7}^{n+1} \\
& - \widehat{a}_{4y} \left[2\widehat{a}_{1y} + \widehat{b}_{1,8} + \widehat{b}_{1,9} \right] \Psi_R|_{1,9}^{n+1} + \frac{\widehat{a}_{2y}^2}{2} \Psi_R|_{1,6}^{n+1} \\
& = V_R|_{1,8}^{n+\frac{1}{2}} - \widehat{a}_{2y} \left(V_I|_{1,9}^{n+\frac{1}{2}} + V_I|_{1,7}^{n+\frac{1}{2}} \right) + (\widehat{a}_{1y} + \widehat{b}_{1,8}) V_I|_{1,8}^{n+\frac{1}{2}} - \frac{\widehat{a}_{2y}^2}{2} \Psi_R|_{1,10}^{n+1}
\end{aligned} \tag{5.4.4c}$$

$$\begin{aligned}
& \frac{1}{2} \left[1 + \widehat{a}_{2y}^2 + (\widehat{a}_{1y} + \widehat{b}_{1,9})^2 \right] \Psi_R|_{1,9}^{n+1} - \widehat{a}_{4y} \left[2\widehat{a}_{1y} + \widehat{b}_{1,8} + \widehat{b}_{1,9} \right] \Psi_R|_{1,8}^{n+1} + \frac{\widehat{a}_{2y}^2}{2} \Psi_R|_{1,7}^{n+1} \\
& = V_R|_{1,9}^{n+\frac{1}{2}} - \widehat{a}_{2y} V_I|_{1,8}^{n+\frac{1}{2}} + (\widehat{a}_{1y} + \widehat{b}_{1,9}) V_I|_{1,9}^{n+\frac{1}{2}} + \widehat{a}_{4y} (\widehat{a}_{1y} + \widehat{b}_{1,9}) \Psi_R|_{1,10}^{n+1} - \widehat{a}_{4y} \Psi_I|_{1,10}^{n+1}.
\end{aligned} \tag{5.4.4d}$$

5.5 Numerical Experiments

In this section, we compare the numerical errors of the Penta-FADI method with two examples where the exact solutions are known. Next, we investigate the efficiency of the Tri-FADI, Penta-ADI and Penta-FADI methods compared to the conventional Tri-ADI method.

5.5.1 Example 1

The exact solution of a transient Gaussian distribution is given as [164]

$$\Psi_{\text{exact}}(x, y, t) = \frac{\mathbf{j}\sqrt{2}}{(\mathbf{j} - 4t)\sqrt{\pi}} e^{-\mathbf{j}((x^2+y^2+\mathbf{j}k_0x+\mathbf{j}k_0^2t)/(\mathbf{j}-4t))} \tag{5.5.1}$$

which has a zero potential function

$$\widehat{\omega}(x, y) = 0. \tag{5.5.2}$$

Figure 5.2: Waveform at observation point (31,51) for Example 1.

The exact solution in (5.5.1) is normalized such that the conservation of probability function ($|\Psi|^2$) of the entire domain is satisfied. Note that for the numerical solution of this exact solution, we set $\hbar = 1$ and $m_e = 0.5$. We are able to obtain the initial condition in (2.3.7) by setting $t = 0$ s in (5.5.1):

$$\phi_R(x, y) = \Psi_{R_{\text{exact}}}(x, y, 0), \quad (5.5.3)$$

$$\phi_I(x, y) = \Psi_{I_{\text{exact}}}(x, y, 0) \quad (5.5.4)$$

and the Dirichlet boundary condition with $\Lambda = 0$ in (2.3.8):

$$f_R(a, b, t) = \Psi_{R_{\text{exact}}}(a, b, t) \quad (5.5.5)$$

$$f_I(a, b, t) = \Psi_{R_{\text{exact}}}(a, b, t) \quad (5.5.6)$$

Table 5.1: Maximum relative error for the Penta-FADI method at different times for Example 1 with $\Delta x = \Delta y = 0.05$ m and $\Delta t = 6.25 \times 10^{-4}$ s.

t	Maximum Relative Error					
	Penta-FADI (with Δt)		Penta-FADI (with $20\Delta t$)		Penta-FADI (with $40\Delta t$)	
	real	imag	real	imag	real	imag
0.1	2.8×10^{-3}	3.2×10^{-3}	6.2×10^{-3}	6.9×10^{-3}	1.6×10^{-2}	1.8×10^{-2}
0.3	6.5×10^{-3}	1.5×10^{-2}	1.5×10^{-2}	3.5×10^{-2}	4.5×10^{-2}	9.6×10^{-2}
0.5	1.1×10^{-2}	1.3×10^{-2}	2.3×10^{-2}	2.2×10^{-2}	6.6×10^{-2}	6.1×10^{-2}
0.7	1.4×10^{-2}	1.4×10^{-2}	3.0×10^{-2}	3.6×10^{-2}	8.6×10^{-2}	8.1×10^{-2}
1.0	3.3×10^{-2}	4.4×10^{-2}	6.3×10^{-2}	9.1×10^{-2}	1.5×10^{-1}	2.7×10^{-1}

where

$$f(x, y, t) = f_R(x, y, t) + \mathbf{j}f_I(a, b, t), \quad \phi(x, y) = \phi_R(x, y) + \mathbf{j}\phi_I(x, y),$$

$a = 0$ or 5 , $b = 0$ or 5 and $0 < t \leq T$.

As this is a open boundary problem, we limit the computation domain as $0 \leq x \leq 5$, $0 \leq y \leq 5$, $T = 1$ s, $\Delta x = \Delta y = 0.05$ m and $\Delta t = 6.25 \times 10^{-4}$ s. The Gaussian wave starts from the center of the domain $x = y = 2.5$ m with a wave number of $k_0 = 2.5$. The maximum relative error of the Penta-FADI method is computed as

$$\text{Error} = \frac{\max_{\forall i, \forall j} \sqrt{|\Psi_{\text{exact}}|_{i,j} - \Psi_{\text{Penta-FADI}}|_{i,j}|^2}}{\max_{\forall i, \forall j} \sqrt{|\Psi_{\text{exact}}|_{i,j}|^2}}. \quad (5.5.7)$$

Figure 5.2 shows the waveform obtained from $|\Psi|$ with different Δt at observation point (31,51). It is worth noting that all three curves of the Penta-FADI method are very close to the exact solution. This proves that a good accuracy can be obtained through the Penta-FADI method.

Table 5.2: Maximum relative error for the Penta-FADI method at time $T = 1$ s for Example 1 with different Δx and Δy where $\Delta t = 6.25 \times 10^{-4}$ s.

$\Delta x = \Delta y$	Maximum Relative Error		
	Penta-FADI (with Δt)	Penta-FADI (with $20\Delta t$)	Penta-FADI (with $40\Delta t$)
	absolute	absolute	absolute
0.05	4.1×10^{-2}	8.3×10^{-2}	2.3×10^{-1}
0.025	1.0×10^{-2}	5.0×10^{-2}	1.9×10^{-1}
0.0125	2.6×10^{-3}	4.2×10^{-2}	1.8×10^{-1}

Table 5.1 reports the maximum relative error for the Penta-FADI method in both real and imaginary parts of the wave function Ψ for Δt , $20\Delta t$ and $40\Delta t$. It can be seen that as Δt increases, the error for the Penta-FADI method increases. This is inevitable as the increase in Δt will incur a greater error but this may still be tolerable for some applications.

We further investigate the accuracy of the Penta-FADI method by reducing the spatial steps Δx and Δy . Table 5.2 shows the maximum relative error for the Penta-FADI method at time $T = 1$ s at finer Δx and Δy . As can be seen from the Table, we can deduce that finer spatial step would result in smaller error.

Next, we look into the conservation of probability for the Schrödinger equation. We extend the computation domain to $0 \leq x \leq 40$, $0 \leq y \leq 40$ and the Gaussian wave starts from the center of the domain $x = y = 20$ m. This will enable the computation domain to contain the whole Gaussian wave for the entire simulation time. Figure 5.3 shows the probability function ($|\Psi|^2$) for the Penta-FADI method with various Δt in Example 1, where the probability obtained is exactly 1. This justifies the conservation of probability for the Schrödinger equation.

Figure 5.3: Probability of Example 1.

5.5.2 Example 2

The exact solution of this example [91]

$$\Psi_{\text{exact}}(x, y, t) = \frac{5}{50^5} x^2 y^2 e^{it} \quad (5.5.8)$$

comes with a potential function

$$\hat{\omega}(x, y) = \frac{2}{x^2} + \frac{2}{y^2} - 1 \quad (5.5.9)$$

where the initial and boundary conditions are similar to (5.5.3)-(5.5.6). The exact solution in (5.5.8) has been normalized to satisfy the conservation of probability function ($|\Psi|^2$) of the entire domain which will be investigated below. Note that for the numerical solution of this exact solution, we set $\hbar = 1$ and $m_e = 0.5$.

The computation domain is given as $0 < x \leq 50$, $0 < y \leq 50$, $T = 1$ s, $\Delta x = \Delta y = 0.1$ m and $\Delta t = 1.0 \times 10^{-3}$ s. Figures 5.4 and 5.5 show the waveforms for real and

Figure 5.4: Waveform of $\Psi_R|_{250,250}$ for Example 2. Figure 5.5: Waveform of $\Psi_I|_{250,250}$ for Example 2.

Table 5.3: Maximum relative error for the Penta-FADI method at different times for Example 2 with $\Delta x = \Delta y = 0.1$ m and $\Delta t = 1.0 \times 10^{-3}$ s.

t	Maximum Relative Error					
	Penta-FADI (with Δt)		Penta-FADI (with $50\Delta t$)		Penta-FADI (with $100\Delta t$)	
	real	imag	real	imag	real	imag
0.1	1.0×10^{-9}	2.6×10^{-8}	2.6×10^{-6}	6.1×10^{-5}	8.9×10^{-6}	2.3×10^{-4}
0.3	2.9×10^{-9}	2.5×10^{-8}	7.5×10^{-6}	6.4×10^{-5}	2.5×10^{-5}	2.5×10^{-4}
0.5	7.1×10^{-9}	2.4×10^{-8}	1.7×10^{-5}	6.1×10^{-5}	7.3×10^{-5}	2.5×10^{-4}
0.7	1.4×10^{-8}	2.4×10^{-8}	3.5×10^{-5}	5.9×10^{-5}	1.4×10^{-4}	2.3×10^{-4}
1.0	3.5×10^{-8}	2.2×10^{-8}	8.7×10^{-5}	5.5×10^{-5}	3.5×10^{-4}	2.2×10^{-4}

Table 5.4: CPU time and relative maximum error of the Penta-FADI method with $\Delta x = \Delta y = 0.1$ m and $\Delta t = 1.0 \times 10^{-3}$ s.

At t=1.0s	Δt	$50\Delta t$	$100\Delta t$
Tri-ADI	1657.97s	32.97s	16.50s
Penta-FADI	83.86s	1.69s	0.86s
Eff. Gain	19.8	19.5	19.2
Relative Max. Error Penta-FADI	2.32×10^{-8}	5.81×10^{-5}	2.34×10^{-4}

imaginary parts of Ψ at observation point (250,250). It can be seen that all three curves of the Penta-FADI method are very close to the exact solution. This again shows that the results of the Penta-FADI method as accurate approximations of the Schrödinger equation.

Table 5.3 reports the maximum relative error for the Penta-FADI method in both real and imaginary parts of the wave function Ψ for Δt , $50\Delta t$ and $100\Delta t$. The CPU time and relative maximum error are further shown in Table 5.4. The numerical experiment of Example 2 is simulated using Microsoft Visual C++ codes with Intel(R) Xeon(R) 2.40 GHz CPU. We find that the efficiency gain of the Penta-FADI

Figure 5.6: Probability of Example 2.

method over Tri-ADI method for various Δt is approximately 19 times. It can be seen that as the Δt increases, the relative maximum error increases. At $50\Delta t$, the Penta-FADI method allows us to choose Δt to be 50 times larger at the expense of 5.81×10^{-5} relative maximum error. Even at $100\Delta t$, the Penta-FADI method allows us to choose Δt to be 100 times larger at the expense of 2.34×10^{-4} relative maximum error. This shows that the Penta-FADI method exhibits good trade-off between accuracy and efficiency.

Next, we investigate the conservation of probability for the Schrödinger equation. Figure 5.6 shows the probability function ($|\Psi|^2$) for the Penta-FADI method with various Δt in Example 2. It can be seen that the curve deviates as Δt increases. Furthermore, due to the discretization of the computation domain into grids, the probability achieved is 0.99 which is close to the actual value of 1.

Table 5.5: Efficiency gains of Tri- and Penta-FADI methods over Tri-ADI method for various computation domains ($\Delta t = 4.0 \times 10^{-3}$ s, 250 time steps)

Domain Size	Efficiency Gain		
	Tri-FADI	Penta-ADI	Penta-FADI
	v.s. Tri-ADI	v.s. Tri-ADI	v.s. Tri-ADI
1000×1000	1.26	24.63	33.27
2000×2000	1.26	24.14	33.61
3000×3000	1.26	23.74	32.07
4000×4000	1.26	22.41	30.16

5.5.3 Efficiency

To further validate the efficiency gain of our proposed Penta-FADI and Tri-FADI methods over conventional Tri-ADI method, the computation domain size ranges from 1000×1000 to 4000×4000 grids, with $\Delta t = 4.0 \times 10^{-3}$ s over 250 time steps have been considered. The CPU computation time of the Penta-ADI, Penta-FADI and Tri-ADI methods is obtained and the efficiency gains are tabulated.

Table 5.5 shows the efficiency gains among various methods over a range of computation domains. Comparison between the Tri-FADI method with the Tri-ADI method shows an efficiency gain of 1.26 times. This verifies the reduction of the arithmetic operators on the RHS of (5.3.22). The efficiency gain is low because complex variables are still present for the Tri-FADI method. The efficiency gain of the Penta-ADI method over Tri-ADI method is over 22 times. This can be explained by the complex wave function Ψ in the Tri-ADI method, where the direct computation of complex variables is rather complicated, leading to longer CPU computation time. By further comparing the Penta-FADI method with the Tri-ADI method, the efficiency gain is over 30 times. This justifies the overall reduction of flops count and

memory indexing of the Penta-FADI method (as mentioned in Section 5.3). This is because it has matrix-operator-free RHS, leading to reduction in the number of arithmetic operations to the minimal.

5.6 Conclusions

We have presented a novel Penta-ADI method for the 2-D Schrödinger equation. Through the separation of the complex wave function into real and imaginary parts, a pentadiagonal system of equations for the ADI method is obtained, which results in our Penta-ADI method. The Penta-ADI method has been further simplified into the Penta-FADI method with matrix-operator-free RHS, leading to computationally efficient update equations. As the Penta-FADI method involves five stencils on the LHS of pentadiagonal update equations, special treatments that are required for the implementation of the Dirichlet boundary condition have been discussed. Using the Penta-FADI method, a high efficiency gain can be achieved over the Tri-ADI method, which involves computations of the complex wave function.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis has focused on the FDTD methods on three specific areas, namely, electromagnetics, heat transfer (thermodynamics) and Schrödinger equation (quantum mechanics). The research contributions for each specific area in this thesis are concluded as follows.

In electromagnetics, the FDTD methods are used to solve Maxwell's equations (which are hyperbolic partial differential equations). Various boundary conditions have been studied and incorporated into the FADI- and FLOD-FDTD methods. For closed region simulation, the PMC and PEC boundary conditions have been derived and implemented into FADI- and FLOD-FDTD methods using image theory. Image theory has been demonstrated to facilitate the implementations of PMC and/or PEC boundary conditions regardless of whether the update equations for electric or magnetic fields are implicit. On the other hand, in open structure simulation, the Mur ABC and PMLs have been incorporated into FADI- and FLOD-FDTD methods. For simple and reasonable absorption (in terms of reflection coefficient), the Mur ABC has been implemented into FADI- and FLOD-FDTD methods using consistent implementation and a novel implementation with lower reflection coefficient.

The reflection coefficients for both implementations in FADI- and FLOD-FDTD methods have been compared and validated with the conventional ADI- and LOD-FDTD methods. Furthermore, the split-field PML and CFS-CPML involving the employment of nonphysical media have been implemented into the FADI-FDTD method. The conventional ADI-FDTD method with PMLs is cast into the compact matrix form and formulated into the FADI-FDTD method with its RHS free of matrix operators. With that, it has reduced the complexity of the PML implementation leading to simpler and more concise update equations. Furthermore, by using CUDA, the FADI-FDTD method with CFS-CPML has been incorporated into the GPU to exploit data parallelism to achieve a high efficiency gain. To demonstrate the usefulness of the FADI-FDTD method with CFS-CPML, a practical microstrip low-pass filter has been presented. A high computational power has been attained while preserving a good agreement with the Yee-FDTD method.

Based on the FDTD technique used for solving Maxwell's equations in electromagnetics, it has been extended to solve the heat transfer equation (which is a parabolic partial differential equation) in thermodynamics. For heat transfer, a proposed stabilized 2-D DG-ADI method has been introduced to alleviate the potential instability caused by the gradient terms for the heat transfer equation within inhomogeneous media. The algorithm is then cast into the (stabilized) PR-ADI method in compact form, and further formulated into the stable FADI method with operator-free RHS. Subsequently, the GPU-accelerated 2-D FADI method of ICs with microchannel cooling has been presented. The convection heat flux due to fluid motion has been included in the heat transfer equation. The cooling effect of microchannels has been verified to maintain the moderate temperature level across IC chips. Also, a 3-D efficient ADI method in homogeneous media has been proposed to overcome the high complexity and memory variables required for the conventional 3-D DG-ADI method. Using the efficient ADI method, the RHS of the first procedure has

been formulated to use only a single operator and heat generation input. It has been demonstrated that there is a decrease in the number of memory variables required, and hence reducing the memory space and memory indexing overhead. The potential instability caused by gradient terms for the 2-D DG-ADI method within inhomogeneous media can be circumvented using the proposed technique, however, it is not equally applicable for the 3-D DG-ADI method. To overcome the potential instability of the 3-D DG-ADI method within inhomogeneous media, two stable 3-D FLOD methods have been presented to solve the heat transfer equation. The stability of the proposed stabilized technique for 2-D FADI and 3-D FLOD methods has been substantiated by the stability analysis through analyzing the eigenvalues of the amplification matrix. To show the effectiveness of the proposed stabilized 3-D FLOD method, the heat distribution of the closely resembled Alpha 21364 processor chip has been presented and analyzed.

By using the FDTD technique to solve the heat transfer equation, this technique has been extended to solve the Schrödinger equation (which is a parabolic partial differential equation with complex variables) in quantum mechanics. A novel Penta-FADI method has been proposed for the 2-D Schrödinger equation. Through the separation of the complex wave function into real and imaginary parts, a Penta-ADI method has been obtained, which is further simplified into the Penta-FADI method with matrix-operator-free RHS, leading to computationally efficient update equations. As the Penta-FADI method involves five stencils on the LHS of pentadiagonal update equations, special treatments that are required for the implementation of the Dirichlet boundary condition have been discussed. Numerical results have justified that the high efficiency gain is achievable for the Penta-FADI method over Tri-ADI method, which involves direct computations of the complex wave function.

As a final conclusion, this thesis has accomplished in the development and extensions

of stable and efficient ADI/LOD FDTD methods for electromagnetics, heat transfer (thermodynamics) and Schrödinger equation (quantum mechanics).

6.2 Future Work

With the development of the efficient fundamental FDTD methods for electromagnetics, they can be readily applied in a lot of interesting engineering and science areas. In biophotonics area for instance, simulation of normal cell and virus/cancerous cell can be carried out for the purpose of detection and cure of terminal illness. The increasingly popular topics of photonic crystal structures [165–168], metamaterials and plasmonic waveguides [169] can also be ventured into with the help of the efficient fundamental FDTD methods.

The development for an analytical proof of unconditionally stability of DG-ADI FDTD and LOD-FDTD methods for the heat transfer equation for inhomogeneous media is recommended. Furthermore, with the highly efficient FADI-FDTD method introduced for the Schrödinger equation, it is of great interest to explore the parallel computing using the GPU to boost computational efficiency.

As the size of devices is approaching nanoscales and beyond, research areas such as nano-photonics, plasmonics and nano-electronics have attracted special attention. Quantum effects which affect data transfer speed and etc. cannot be neglected. A hybrid numerical approach to simulate the coupled Maxwell's and Schrödinger equations [170, 171] can be introduced. This hybrid numerical simulation can be used to analyze the quantum effect for carbon nano-tube [172, 173], tunneling current through potential barriers [174], etc. The use of our efficient fundamental FDTD methods can greatly enhance the hybrid numerical approach for the coupled Maxwell's and Schrödinger equations.

Appendix A

Mur ABC for Yee's explicit FDTD Method

For the Yee's explicit FDTD method, implementing the Mur first order ABC in a computation domain of $ie \times je \times ke$ grids, where the grids are indexed from 0 to ie , 0 to je and 0 to ke in the x -, y - and z -directions respectively, will result in the following:

$$E_x|_{i+\frac{1}{2},0,k}^{n+1} = E_x|_{i+\frac{1}{2},1,k}^n + \frac{v\Delta t - \Delta y}{v\Delta t + \Delta y} \left(E_x|_{i+\frac{1}{2},1,k}^{n+1} - E_x|_{i+\frac{1}{2},0,k}^n \right) \quad (6.2.1a)$$

$$E_x|_{i+\frac{1}{2},j,0}^{n+1} = E_x|_{i+\frac{1}{2},j,1}^n + \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z} \left(E_x|_{i+\frac{1}{2},j,1}^{n+1} - E_x|_{i+\frac{1}{2},j,0}^n \right) \quad (6.2.1b)$$

$$E_y|_{0,j+\frac{1}{2},k}^{n+1} = E_y|_{1,j+\frac{1}{2},k}^n + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} \left(E_y|_{1,j+\frac{1}{2},k}^{n+1} - E_y|_{0,j+\frac{1}{2},k}^n \right) \quad (6.2.1c)$$

$$E_y|_{i,j+\frac{1}{2},0}^{n+1} = E_y|_{i,j+\frac{1}{2},1}^n + \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z} \left(E_y|_{i,j+\frac{1}{2},1}^{n+1} - E_y|_{i,j+\frac{1}{2},0}^n \right) \quad (6.2.1d)$$

$$E_z|_{0,j,k+\frac{1}{2}}^{n+1} = E_z|_{1,j,k+\frac{1}{2}}^n + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} \left(E_z|_{1,j,k+\frac{1}{2}}^{n+1} - E_z|_{0,j,k+\frac{1}{2}}^n \right) \quad (6.2.1e)$$

$$E_z|_{i,0,k+\frac{1}{2}}^{n+1} = E_z|_{i,1,k+\frac{1}{2}}^n + \frac{v\Delta t - \Delta y}{v\Delta t + \Delta y} \left(E_z|_{i,1,k+\frac{1}{2}}^{n+1} - E_z|_{i,0,k+\frac{1}{2}}^n \right) \quad (6.2.1f)$$

$$E_x|_{i+\frac{1}{2},je,k}^{n+1} = E_x|_{i+\frac{1}{2},jd,k}^n + \frac{v\Delta t - \Delta y}{v\Delta t + \Delta y} \left(E_x|_{i+\frac{1}{2},jd,k}^{n+1} - E_x|_{i+\frac{1}{2},je,k}^n \right) \quad (6.2.1g)$$

$$E_x|_{i+\frac{1}{2},j,ke}^{n+1} = E_x|_{i+\frac{1}{2},j,kd}^n + \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z} \left(E_x|_{i+\frac{1}{2},j,kd}^{n+1} - E_x|_{i+\frac{1}{2},j,ke}^n \right) \quad (6.2.1h)$$

$$E_y|_{ie,j+\frac{1}{2},k}^{n+1} = E_y|_{id,j+\frac{1}{2},k}^n + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} \left(E_y|_{id,j+\frac{1}{2},k}^{n+1} - E_y|_{ie,j+\frac{1}{2},k}^n \right) \quad (6.2.1i)$$

$$E_y|_{i,j+\frac{1}{2},ke}^{n+1} = E_y|_{i,j+\frac{1}{2},kd}^n + \frac{v\Delta t - \Delta z}{v\Delta t + \Delta z} \left(E_y|_{i,j+\frac{1}{2},kd}^{n+1} - E_y|_{i,j+\frac{1}{2},ke}^n \right) \quad (6.2.1j)$$

$$E_z|_{ie,j,k+\frac{1}{2}}^{n+1} = E_z|_{id,j,k+\frac{1}{2}}^n + \frac{v\Delta t - \Delta x}{v\Delta t + \Delta x} \left(E_z|_{id,j,k+\frac{1}{2}}^{n+1} - E_z|_{ie,j,k+\frac{1}{2}}^n \right) \quad (6.2.1k)$$

$$E_z|_{i,je,k+\frac{1}{2}}^{n+1} = E_z|_{i,jd,k+\frac{1}{2}}^n + \frac{v\Delta t - \Delta y}{v\Delta t + \Delta y} \left(E_z|_{i,jd,k+\frac{1}{2}}^{n+1} - E_z|_{i,je,k+\frac{1}{2}}^n \right) \quad (6.2.1l)$$

where

$$id = ie - 1, \quad jd = je - 1, \quad kd = ke - 1.$$

The update equations of the Mur ABC are relatively simple with low memory consumption. It has a reasonable reflection coefficient which should be adequate for practical use.

Appendix B

CFS-CPML for Yee's explicit FDTD Method

For the full update equations of the Yee's explicit FDTD method with CFS-CPML, expanding (2.1.33), upon some manipulations and arrangements will result in the following:

$$\begin{aligned}
 E_x|_{i+\frac{1}{2},j,k}^{n+1} = & E_x|_{i+\frac{1}{2},j,k}^n \\
 & + \frac{\Delta t}{\epsilon} \left(\frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}}}{\widehat{\kappa}_{y_j} \Delta y} - \frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}}}{\widehat{\kappa}_{z_k} \Delta z} \right. \\
 & \left. + \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.2a)
 \end{aligned}$$

$$\begin{aligned}
 E_y|_{i,j+\frac{1}{2},k}^{n+1} = & E_y|_{i,j+\frac{1}{2},k}^n \\
 & + \frac{\Delta t}{\epsilon} \left(\frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}}}{\widehat{\kappa}_{z_k} \Delta z} - \frac{H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}}}{\widehat{\kappa}_{x_i} \Delta x} \right. \\
 & \left. + \psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.2b)
 \end{aligned}$$

$$\begin{aligned}
 E_z|_{i,j,k+\frac{1}{2}}^{n+1} = & E_z|_{i,j,k+\frac{1}{2}}^n \\
 & + \frac{\Delta t}{\epsilon} \left(\frac{H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}}}{\widehat{\kappa}_{x_i} \Delta x} - \frac{H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}}}{\widehat{\kappa}_{y_j} \Delta y} \right. \\
 & \left. + \psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.2c)
 \end{aligned}$$

$$\begin{aligned}
 H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{3}{2}} = & H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \\
 & + \frac{\Delta t}{\mu} \left(\frac{E_y|_{i,j+\frac{1}{2},k+1}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}} \Delta z} - \frac{E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}} \Delta y} \right)
 \end{aligned}$$

$$+ \psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} \Big) \quad (6.2.2d)$$

$$\begin{aligned} H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{3}{2}} = & H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\ & + \frac{\Delta t}{\mu} \left(\frac{E_z|_{i+1,j,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}\Delta x} - \frac{E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}}\Delta z} \right. \\ & \left. + \psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} \right) \end{aligned} \quad (6.2.2e)$$

$$\begin{aligned} H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{3}{2}} = & H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \\ & + \frac{\Delta t}{\mu} \left(\frac{E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}\Delta y} - \frac{E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}\Delta x} \right. \\ & \left. + \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} \right) \end{aligned} \quad (6.2.2f)$$

Here, ψ_e and ψ_h are discrete variables with nonzero values only in their corresponding PML regions. The update equations of these variables are

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \widehat{C}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} + \frac{\widehat{b}_{y_j}}{\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.3a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \widehat{C}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} + \frac{\widehat{b}_{z_k}}{\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.3b)$$

$$\psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \widehat{C}_{x_i} \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n-\frac{1}{2}} + \frac{\widehat{b}_{x_i}}{\Delta x} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.3c)$$

$$\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \widehat{C}_{z_k} \psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n-\frac{1}{2}} + \frac{\widehat{b}_{z_k}}{\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.3d)$$

$$\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \widehat{C}_{x_i} \psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{\widehat{b}_{x_i}}{\Delta x} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.3e)$$

$$\psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \widehat{C}_{y_j} \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{\widehat{b}_{y_j}}{\Delta y} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.3f)$$

$$\psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \widehat{C}_{y_{j+\frac{1}{2}}} \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\widehat{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.3g)$$

$$\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \widehat{C}_{z_{k+\frac{1}{2}}} \psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\widehat{b}_{z_{k+\frac{1}{2}}}}{\Delta z} \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.3h)$$

$$\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = \widehat{C}_{x_{i+\frac{1}{2}}} \psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\widehat{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.3i)$$

$$\psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = \widehat{C}_{z_{k+\frac{1}{2}}} \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\widehat{b}_{z_{k+\frac{1}{2}}}}{\Delta z} \left(E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.3j)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \widehat{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\widehat{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.3k)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \widehat{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\widehat{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.3l)$$

From the equations above, two discrete variables are required for each field point. The CFS-CPML is able to absorb evanescent waves and signals of long time-signature effectively. However, it comes with an expense of high complexity and high computer memory consumption.

Appendix C

Mur ABC for ADI- and LOD-FDTD Methods

For the conventional 3-D ADI- and LOD-FDTD algorithms, implementing the Mur first order ABC in a computation domain of $ie \times je \times ke$ grids, where the grids are indexed from 0 to ie , 0 to je and 0 to ke in the x -, y - and z -directions respectively, will result in the following:

i) Consistent Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \\ = \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^n + \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^n \end{aligned} \quad (6.2.4a)$$

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} \\ = \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},0}^n + \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},1}^n \end{aligned} \quad (6.2.4b)$$

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\ = \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_z|_{0,j,k+\frac{1}{2}}^n + \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_z|_{1,j,k+\frac{1}{2}}^n \end{aligned} \quad (6.2.4c)$$

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} \\ = \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},je,k}^n + \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},jd,k}^n \end{aligned} \quad (6.2.4d)$$

$$\begin{aligned} \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} \\ = \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},ke}^n + \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},kd}^n \end{aligned} \quad (6.2.4e)$$

$$\left(1 + \frac{v\Delta t}{2\Delta x}\right) E_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}}$$

$$= \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_z|_{ie,j,k+\frac{1}{2}}^n + \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_z|_{id,j,k+\frac{1}{2}}^n \quad (6.2.4f)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},j,1}^n + \frac{2\Delta z - v\Delta t}{2\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,0}^n - E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \right) \quad (6.2.4g)$$

$$E_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} = E_y|_{1,j+\frac{1}{2},k}^n + \frac{2\Delta x - v\Delta t}{2\Delta x + v\Delta t} \left(E_y|_{0,j+\frac{1}{2},k}^n - E_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.4h)$$

$$E_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} = E_z|_{i,1,k+\frac{1}{2}}^n + \frac{2\Delta y - v\Delta t}{2\Delta y + v\Delta t} \left(E_z|_{i,0,k+\frac{1}{2}}^n - E_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.4i)$$

$$E_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},j,kd}^n + \frac{2\Delta z - v\Delta t}{2\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,ke}^n - E_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} \right) \quad (6.2.4j)$$

$$E_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} = E_y|_{id,j+\frac{1}{2},k}^n + \frac{2\Delta x - v\Delta t}{2\Delta x + v\Delta t} \left(E_y|_{ie,j+\frac{1}{2},k}^n - E_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.4k)$$

$$E_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} = E_z|_{i,jd,k+\frac{1}{2}}^n + \frac{2\Delta y - v\Delta t}{2\Delta y + v\Delta t} \left(E_z|_{i,je,k+\frac{1}{2}}^n - E_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.4l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\begin{aligned} & \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+1} \\ &= \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.5a)$$

$$\begin{aligned} & \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_y|_{0,j+\frac{1}{2},k}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_y|_{1,j+\frac{1}{2},k}^{n+1} \\ &= \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.5b)$$

$$\begin{aligned} & \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,0,k+\frac{1}{2}}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,1,k+\frac{1}{2}}^{n+1} \\ &= \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.5c)$$

$$\begin{aligned} & \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,ke}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,kd}^{n+1} \\ &= \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.5d)$$

$$\begin{aligned} & \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_y|_{ie,j+\frac{1}{2},k}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_y|_{id,j+\frac{1}{2},k}^{n+1} \\ &= \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.5e)$$

$$\begin{aligned} & \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,je,k+\frac{1}{2}}^{n+1} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,jd,k+\frac{1}{2}}^{n+1} \\ &= \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.5f)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},0,k}^{n+1} = E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} + \frac{2\Delta y - v\Delta t}{2\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},1,k}^{n+1} \right) \quad (6.2.5g)$$

$$E_y|_{i,j+\frac{1}{2},0}^{n+1} = E_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} + \frac{2\Delta z - v\Delta t}{2\Delta z + v\Delta t} \left(E_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},1}^{n+1} \right) \quad (6.2.5h)$$

$$E_z|_{0,j,k+\frac{1}{2}}^{n+1} = E_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{2\Delta x - v\Delta t}{2\Delta x + v\Delta t} \left(E_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{1,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.5i)$$

$$E_x|_{i+\frac{1}{2},je,k}^{n+1} = E_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} + \frac{2\Delta y - v\Delta t}{2\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},jd,k}^{n+1} \right) \quad (6.2.5j)$$

$$E_y|_{i,j+\frac{1}{2},ke}^{n+1} = E_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} + \frac{2\Delta z - v\Delta t}{2\Delta z + v\Delta t} \left(E_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},kd}^{n+1} \right) \quad (6.2.5k)$$

$$E_z|_{ie,j,k+\frac{1}{2}}^{n+1} = E_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{2\Delta x - v\Delta t}{2\Delta x + v\Delta t} \left(E_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{id,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.5l)$$

ii) Novel Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},0,k}^n + E_x|_{i+\frac{1}{2},1,k}^n \quad (6.2.6a)$$

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) E_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta z}\right) E_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} = E_y|_{i,j+\frac{1}{2},0}^n + E_y|_{i,j+\frac{1}{2},1}^n \quad (6.2.6b)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) E_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta x}\right) E_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = E_z|_{0,j,k+\frac{1}{2}}^n + E_z|_{1,j,k+\frac{1}{2}}^n \quad (6.2.6c)$$

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) E_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta y}\right) E_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},je,k}^n + E_x|_{i+\frac{1}{2},jd,k}^n \quad (6.2.6d)$$

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) E_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta z}\right) E_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} = E_y|_{i,j+\frac{1}{2},ke}^n + E_y|_{i,j+\frac{1}{2},kd}^n \quad (6.2.6e)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) E_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta x}\right) E_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = E_z|_{ie,j,k+\frac{1}{2}}^n + E_z|_{id,j,k+\frac{1}{2}}^n \quad (6.2.6f)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{\Delta z}{\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,0}^n + E_x|_{i+\frac{1}{2},j,1}^n \right) \quad (6.2.6g)$$

$$E_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} E_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta x}{\Delta x + v\Delta t} \left(E_y|_{0,j+\frac{1}{2},k}^n + E_x|_{1,j+\frac{1}{2},k}^n \right) \quad (6.2.6h)$$

$$E_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} E_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta y}{\Delta y + v\Delta t} \left(E_z|_{i,0,k+\frac{1}{2}}^n + E_z|_{i,1,k+\frac{1}{2}}^n \right) \quad (6.2.6i)$$

$$E_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} + \frac{\Delta z}{\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,ke}^n + E_x|_{i+\frac{1}{2},j,kd}^n \right) \quad (6.2.6j)$$

$$E_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} E_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta x}{\Delta x + v\Delta t} \left(E_y|_{ie,j+\frac{1}{2},k}^n + E_y|_{id,j+\frac{1}{2},k}^n \right) \quad (6.2.6k)$$

$$E_z|_{i,j,e,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} E_z|_{i,j,d,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta y}{\Delta y + v\Delta t} \left(E_z|_{i,j,e,k+\frac{1}{2}}^n + E_z|_{i,j,d,k+\frac{1}{2}}^n \right) \quad (6.2.6l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+1} = E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (6.2.7a)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) E_y|_{0,j+\frac{1}{2},k}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta x}\right) E_y|_{1,j+\frac{1}{2},k}^{n+1} = E_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} + E_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (6.2.7b)$$

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) E_z|_{i,0,k+\frac{1}{2}}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta y}\right) E_z|_{i,1,k+\frac{1}{2}}^{n+1} = E_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.7c)$$

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) E_x|_{i+\frac{1}{2},j,ke}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta z}\right) E_x|_{i+\frac{1}{2},j,kd}^{n+1} = E_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} \quad (6.2.7d)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) E_y|_{ie,j+\frac{1}{2},k}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta x}\right) E_y|_{id,j+\frac{1}{2},k}^{n+1} = E_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} + E_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (6.2.7e)$$

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) E_z|_{i,j,e,k+\frac{1}{2}}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta y}\right) E_z|_{i,j,d,k+\frac{1}{2}}^{n+1} = E_z|_{i,j,e,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,j,d,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.7f)$$

b) Explicit Update

$$E_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{\Delta y}{\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \right) \quad (6.2.7g)$$

$$E_y|_{i,j+\frac{1}{2},0}^{n+1} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} E_y|_{i,j+\frac{1}{2},1}^{n+1} + \frac{\Delta z}{\Delta z + v\Delta t} \left(E_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + E_x|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} \right) \quad (6.2.7h)$$

$$E_z|_{0,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} E_z|_{1,j,k+\frac{1}{2}}^{n+1} + \frac{\Delta x}{\Delta x + v\Delta t} \left(E_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.7i)$$

$$E_x|_{i+\frac{1}{2},je,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},jd,k}^{n+1} + \frac{\Delta y}{\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} \right) \quad (6.2.7j)$$

$$E_y|_{i,j+\frac{1}{2},ke}^{n+1} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} E_y|_{i,j+\frac{1}{2},kd}^{n+1} + \frac{\Delta z}{\Delta z + v\Delta t} \left(E_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + E_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} \right) \quad (6.2.7k)$$

$$E_z|_{ie,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} E_z|_{id,j,k+\frac{1}{2}}^{n+1} + \frac{\Delta x}{\Delta x + v\Delta t} \left(E_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.7l)$$

where

$$id = ie - 1, \quad jd = je - 1, \quad kd = ke - 1.$$

Appendix D

Mur ABC for FADI-FDTD Method

With the formulation of the Mur ABC for both consistent and novel implementations in 3-D FADI-FDTD method, implementing the Mur first order ABC in a computation domain of $ie \times je \times ke$ grids, where the grids are indexed from 0 to ie , 0 to je and 0 to ke in the x -, y - and z -directions respectively, will result in the following:

i) Consistent Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y}\right)\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y}\right)\tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2},0,k}^n + e_x|_{i+\frac{1}{2},1,k}^n \quad (6.2.8a)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z}\right)\tilde{E}_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z}\right)\tilde{E}_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} = e_y|_{i,j+\frac{1}{2},0}^n + e_y|_{i,j+\frac{1}{2},1}^n \quad (6.2.8b)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x}\right)\tilde{E}_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x}\right)\tilde{E}_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = e_z|_{0,j,k+\frac{1}{2}}^n + e_z|_{1,j,k+\frac{1}{2}}^n \quad (6.2.8c)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y}\right)\tilde{E}_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y}\right)\tilde{E}_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} = e_x|_{i+\frac{1}{2},je,k}^n + e_x|_{i+\frac{1}{2},jd,k}^n \quad (6.2.8d)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z}\right)\tilde{E}_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z}\right)\tilde{E}_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} = e_y|_{i,j+\frac{1}{2},ke}^n + e_y|_{i,j+\frac{1}{2},kd}^n \quad (6.2.8e)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x}\right)\tilde{E}_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x}\right)\tilde{E}_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = e_z|_{ie,j,k+\frac{1}{2}}^n + e_z|_{id,j,k+\frac{1}{2}}^n \quad (6.2.8f)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t}\tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{4\Delta z}{2\Delta z + v\Delta t}\left(e_x|_{i+\frac{1}{2},j,0}^n + e_x|_{i+\frac{1}{2},j,1}^n\right) \quad (6.2.8g)$$

$$\tilde{E}_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t}\tilde{E}_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{4\Delta x}{2\Delta x + v\Delta t}\left(e_y|_{0,j+\frac{1}{2},k}^n + e_y|_{1,j+\frac{1}{2},k}^n\right) \quad (6.2.8h)$$

$$\tilde{E}_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t}\tilde{E}_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{4\Delta y}{2\Delta y + v\Delta t}\left(e_z|_{i,0,k+\frac{1}{2}}^n + e_z|_{i,1,k+\frac{1}{2}}^n\right) \quad (6.2.8i)$$

$$\tilde{E}_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(e_x|_{i+\frac{1}{2},j,ke}^n + e_x|_{i+\frac{1}{2},j,kd}^n \right) \quad (6.2.8j)$$

$$\tilde{E}_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t} \tilde{E}_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(e_y|_{ie,j+\frac{1}{2},k}^n + e_y|_{id,j+\frac{1}{2},k}^n \right) \quad (6.2.8k)$$

$$\tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} \tilde{E}_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(e_z|_{i,j,k+\frac{1}{2}}^n + e_z|_{i,jd,k+\frac{1}{2}}^n \right) \quad (6.2.8l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+1} = e_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (6.2.9a)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x} \right) \tilde{E}_y|_{0,j+\frac{1}{2},k}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x} \right) \tilde{E}_y|_{1,j+\frac{1}{2},k}^{n+1} = e_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} + e_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (6.2.9b)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y} \right) \tilde{E}_z|_{i,0,k+\frac{1}{2}}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y} \right) \tilde{E}_z|_{i,1,k+\frac{1}{2}}^{n+1} = e_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.9c)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,ke}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z} \right) \tilde{E}_x|_{i+\frac{1}{2},j,kd}^{n+1} = e_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} \quad (6.2.9d)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x} \right) \tilde{E}_y|_{ie,j+\frac{1}{2},k}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x} \right) \tilde{E}_y|_{id,j+\frac{1}{2},k}^{n+1} = e_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} + e_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (6.2.9e)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y} \right) \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y} \right) \tilde{E}_z|_{i,jd,k+\frac{1}{2}}^{n+1} = e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.9f)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \right) \quad (6.2.9g)$$

$$\tilde{E}_y|_{i,j+\frac{1}{2},0}^{n+1} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} \tilde{E}_y|_{i,j+\frac{1}{2},1}^{n+1} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(e_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} \right) \quad (6.2.9h)$$

$$\tilde{E}_z|_{0,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t} \tilde{E}_z|_{1,j,k+\frac{1}{2}}^{n+1} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(e_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.9i)$$

$$\tilde{E}_x|_{i+\frac{1}{2},je,k}^{n+1} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},jd,k}^{n+1} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(e_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} \right) \quad (6.2.9j)$$

$$\tilde{E}_y|_{i,j+\frac{1}{2},ke}^{n+1} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} \tilde{E}_y|_{i,j+\frac{1}{2},kd}^{n+1} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(e_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} \right) \quad (6.2.9k)$$

$$\tilde{E}_z|_{ie,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t} \tilde{E}_z|_{id,j,k+\frac{1}{2}}^{n+1} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(e_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.9l)$$

ii) Novel Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) \tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta y}\right) \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},0,k}^n + e_x|_{i+\frac{1}{2},1,k}^n + e_x|_{i+\frac{1}{2},1,k}^{n-\frac{1}{2}} \quad (6.2.10a)$$

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) \tilde{E}_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta z}\right) \tilde{E}_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} = \tilde{E}_y|_{i,j+\frac{1}{2},0}^n + e_y|_{i,j+\frac{1}{2},1}^n + e_y|_{i,j+\frac{1}{2},1}^{n-\frac{1}{2}} \quad (6.2.10b)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) \tilde{E}_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta x}\right) \tilde{E}_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{E}_z|_{0,j,k+\frac{1}{2}}^n + e_z|_{1,j,k+\frac{1}{2}}^n + e_z|_{1,j,k+\frac{1}{2}}^{n-\frac{1}{2}} \quad (6.2.10c)$$

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) \tilde{E}_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta y}\right) \tilde{E}_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},je,k}^n + e_x|_{i+\frac{1}{2},jd,k}^n + e_x|_{i+\frac{1}{2},jd,k}^{n-\frac{1}{2}} \quad (6.2.10d)$$

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) \tilde{E}_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta z}\right) \tilde{E}_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} = \tilde{E}_y|_{i,j+\frac{1}{2},ke}^n + e_y|_{i,j+\frac{1}{2},kd}^n + e_y|_{i,j+\frac{1}{2},kd}^{n-\frac{1}{2}} \quad (6.2.10e)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) \tilde{E}_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{\Delta x}\right) \tilde{E}_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{E}_z|_{ie,j,k+\frac{1}{2}}^n + e_z|_{id,j,k+\frac{1}{2}}^n + e_z|_{id,j,k+\frac{1}{2}}^{n-\frac{1}{2}} \quad (6.2.10f)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{\Delta z}{\Delta z + v\Delta t} \left(\tilde{E}_x|_{i+\frac{1}{2},j,0}^n + e_x|_{i+\frac{1}{2},j,1}^n + e_x|_{i+\frac{1}{2},j,1}^{n-\frac{1}{2}} \right) \quad (6.2.10g)$$

$$\tilde{E}_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} \tilde{E}_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta x}{\Delta x + v\Delta t} \left(\tilde{E}_y|_{0,j+\frac{1}{2},k}^n + e_y|_{1,j+\frac{1}{2},k}^n + e_y|_{1,j+\frac{1}{2},k}^{n-\frac{1}{2}} \right) \quad (6.2.10h)$$

$$\tilde{E}_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} \tilde{E}_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta y}{\Delta y + v\Delta t} \left(\tilde{E}_z|_{i,0,k+\frac{1}{2}}^n + e_z|_{i,1,k+\frac{1}{2}}^n + e_z|_{i,1,k+\frac{1}{2}}^{n-\frac{1}{2}} \right) \quad (6.2.10i)$$

$$\tilde{E}_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} + \frac{\Delta z}{\Delta z + v\Delta t} \left(\tilde{E}_x|_{i+\frac{1}{2},j,ke}^n + e_x|_{i+\frac{1}{2},j,kd}^n + e_x|_{i+\frac{1}{2},j,kd}^{n-\frac{1}{2}} \right) \quad (6.2.10j)$$

$$\tilde{E}_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} \tilde{E}_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\Delta x}{\Delta x + v\Delta t} \left(\tilde{E}_y|_{ie,j+\frac{1}{2},k}^n + e_y|_{id,j+\frac{1}{2},k}^n + e_y|_{id,j+\frac{1}{2},k}^{n-\frac{1}{2}} \right) \quad (6.2.10k)$$

$$\tilde{E}_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} \tilde{E}_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\Delta y}{\Delta y + v\Delta t} \left(\tilde{E}_z|_{i,je,k+\frac{1}{2}}^n + e_z|_{i,jd,k+\frac{1}{2}}^n + e_z|_{i,jd,k+\frac{1}{2}}^{n-\frac{1}{2}} \right) \quad (6.2.10l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) \tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta z}\right) \tilde{E}_x|_{i+\frac{1}{2},j,1}^{n+1} = \tilde{E}_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,1}^n \quad (6.2.11a)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) \tilde{E}_y|_{0,j+\frac{1}{2},k}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta x}\right) \tilde{E}_y|_{1,j+\frac{1}{2},k}^{n+1} = \tilde{E}_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} + e_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + e_y|_{1,j+\frac{1}{2},k}^n \quad (6.2.11b)$$

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) \tilde{E}_z|_{i,0,k+\frac{1}{2}}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta y}\right) \tilde{E}_z|_{i,1,k+\frac{1}{2}}^{n+1} = \tilde{E}_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i,1,k+\frac{1}{2}}^n \quad (6.2.11c)$$

$$\left(1 + \frac{v\Delta t}{\Delta z}\right) \tilde{E}_x|_{i+\frac{1}{2},j,ke}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta z}\right) \tilde{E}_x|_{i+\frac{1}{2},j,kd}^{n+1} = \tilde{E}_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},j,kd}^n \quad (6.2.11d)$$

$$\left(1 + \frac{v\Delta t}{\Delta x}\right) \tilde{E}_y|_{ie,j+\frac{1}{2},k}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta x}\right) \tilde{E}_y|_{id,j+\frac{1}{2},k}^{n+1} = \tilde{E}_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} + e_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} + e_y|_{id,j+\frac{1}{2},k}^n \quad (6.2.11e)$$

$$\left(1 + \frac{v\Delta t}{\Delta y}\right) \tilde{E}_z|_{i,je,k+\frac{1}{2}}^{n+1} + \left(1 - \frac{v\Delta t}{\Delta y}\right) \tilde{E}_z|_{i,jd,k+\frac{1}{2}}^{n+1} = \tilde{E}_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{i,jd,k+\frac{1}{2}}^n \quad (6.2.11f)$$

b) Explicit Update

$$\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{\Delta y}{\Delta y + v\Delta t} \left(\tilde{E}_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},1,k}^n \right) \quad (6.2.11g)$$

$$\tilde{E}_y|_{i,j+\frac{1}{2},0}^{n+1} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} \tilde{E}_y|_{i,j+\frac{1}{2},1}^{n+1} + \frac{\Delta z}{\Delta z + v\Delta t} \left(\tilde{E}_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},1}^n \right) \quad (6.2.11h)$$

$$\tilde{E}_z|_{0,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} \tilde{E}_z|_{1,j,k+\frac{1}{2}}^{n+1} + \frac{\Delta x}{\Delta x + v\Delta t} \left(\tilde{E}_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{1,j,k+\frac{1}{2}}^n \right) \quad (6.2.11i)$$

$$\tilde{E}_x|_{i+\frac{1}{2},je,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} \tilde{E}_x|_{i+\frac{1}{2},jd,k}^{n+1} + \frac{\Delta y}{\Delta y + v\Delta t} \left(\tilde{E}_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} + e_x|_{i+\frac{1}{2},jd,k}^n \right) \quad (6.2.11j)$$

$$\tilde{E}_y|_{i,j+\frac{1}{2},ke}^{n+1} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} \tilde{E}_y|_{i,j+\frac{1}{2},kd}^{n+1} + \frac{\Delta z}{\Delta z + v\Delta t} \left(\tilde{E}_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} + e_y|_{i,j+\frac{1}{2},kd}^n \right) \quad (6.2.11k)$$

$$\tilde{E}_z|_{ie,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} \tilde{E}_z|_{id,j,k+\frac{1}{2}}^{n+1} + \frac{\Delta x}{\Delta x + v\Delta t} \left(\tilde{E}_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + e_z|_{id,j,k+\frac{1}{2}}^n \right) \quad (6.2.11l)$$

where

$$id = ie - 1, \quad jd = je - 1, \quad kd = ke - 1.$$

Appendix E

Mur ABC for FLOD-FDTD Method

For 3-D FLOD-FDTD method, the actual update equations of the Mur ABC in a computation domain of $ie \times je \times ke$ grids, where the grids are indexed from 0 to ie , 0 to je and 0 to ke in the x -, y - and z -directions respectively, will result in the following:

i) Consistent Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y}\right)e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y}\right)e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},0,k}^n + E_x|_{i+\frac{1}{2},1,k}^n \quad (6.2.12a)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z}\right)e_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z}\right)e_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} = E_y|_{i,j+\frac{1}{2},0}^n + E_y|_{i,j+\frac{1}{2},1}^n \quad (6.2.12b)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x}\right)e_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x}\right)e_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = E_z|_{0,j,k+\frac{1}{2}}^n + E_z|_{1,j,k+\frac{1}{2}}^n \quad (6.2.12c)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y}\right)e_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y}\right)e_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} = E_x|_{i+\frac{1}{2},je,k}^n + E_x|_{i+\frac{1}{2},jd,k}^n \quad (6.2.12d)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z}\right)e_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z}\right)e_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} = E_y|_{i,j+\frac{1}{2},ke}^n + E_y|_{i,j+\frac{1}{2},kd}^n \quad (6.2.12e)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x}\right)e_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x}\right)e_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = E_z|_{ie,j,k+\frac{1}{2}}^n + E_z|_{id,j,k+\frac{1}{2}}^n \quad (6.2.12f)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t}e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,0}^n + E_x|_{i+\frac{1}{2},j,1}^n \right) \quad (6.2.12g)$$

$$e_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t}e_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(E_y|_{0,j+\frac{1}{2},k}^n + E_y|_{1,j+\frac{1}{2},k}^n \right) \quad (6.2.12h)$$

$$e_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t}e_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(E_z|_{i,0,k+\frac{1}{2}}^n + E_z|_{i,1,k+\frac{1}{2}}^n \right) \quad (6.2.12i)$$

$$e_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} e_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(E_x|_{i+\frac{1}{2},j,ke}^n + E_x|_{i+\frac{1}{2},j,kd}^n \right) \quad (6.2.12j)$$

$$e_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t} e_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(E_y|_{ie,j+\frac{1}{2},k}^n + E_y|_{id,j+\frac{1}{2},k}^n \right) \quad (6.2.12k)$$

$$e_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} e_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(E_z|_{i,je,k+\frac{1}{2}}^n + E_z|_{i,jd,k+\frac{1}{2}}^n \right) \quad (6.2.12l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z} \right) e_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z} \right) e_x|_{i+\frac{1}{2},j,1}^{n+1} = E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \quad (6.2.13a)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x} \right) e_y|_{0,j+\frac{1}{2},k}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x} \right) e_y|_{1,j+\frac{1}{2},k}^{n+1} = E_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} + E_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (6.2.13b)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y} \right) e_z|_{i,0,k+\frac{1}{2}}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y} \right) e_z|_{i,1,k+\frac{1}{2}}^{n+1} = E_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.13c)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta z} \right) e_x|_{i+\frac{1}{2},j,ke}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta z} \right) e_x|_{i+\frac{1}{2},j,kd}^{n+1} = E_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} \quad (6.2.13d)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta x} \right) e_y|_{ie,j+\frac{1}{2},k}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta x} \right) e_y|_{id,j+\frac{1}{2},k}^{n+1} = E_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} + E_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} \quad (6.2.13e)$$

$$\left(\frac{1}{2} + \frac{v\Delta t}{4\Delta y} \right) e_z|_{i,je,k+\frac{1}{2}}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{4\Delta y} \right) e_z|_{i,jd,k+\frac{1}{2}}^{n+1} = E_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.13f)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} e_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \right) \quad (6.2.13g)$$

$$e_y|_{i,j+\frac{1}{2},0}^{n+1} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} e_y|_{i,j+\frac{1}{2},1}^{n+1} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(E_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + E_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} \right) \quad (6.2.13h)$$

$$e_z|_{0,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t} e_z|_{1,j,k+\frac{1}{2}}^{n+1} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(E_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.13i)$$

$$e_x|_{i+\frac{1}{2},je,k}^{n+1} = \frac{v\Delta t - 2\Delta y}{2\Delta y + v\Delta t} e_x|_{i+\frac{1}{2},jd,k}^{n+1} + \frac{4\Delta y}{2\Delta y + v\Delta t} \left(E_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + E_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} \right) \quad (6.2.13j)$$

$$e_y|_{i,j+\frac{1}{2},ke}^{n+1} = \frac{v\Delta t - 2\Delta z}{2\Delta z + v\Delta t} e_y|_{i,j+\frac{1}{2},kd}^{n+1} + \frac{4\Delta z}{2\Delta z + v\Delta t} \left(E_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + E_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} \right) \quad (6.2.13k)$$

$$e_z|_{ie,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - 2\Delta x}{2\Delta x + v\Delta t} e_z|_{id,j,k+\frac{1}{2}}^{n+1} + \frac{4\Delta x}{2\Delta x + v\Delta t} \left(E_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.13l)$$

ii) Novel Implementation

For first procedure from n to $n + \frac{1}{2}$:

a) Implicit Update

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta y}\right) e_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta y}\right) e_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},0,k}^n + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},1,k}^n \end{aligned} \quad (6.2.14a)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta z}\right) e_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta z}\right) e_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},0}^n + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},1}^n \end{aligned} \quad (6.2.14b)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta x}\right) e_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta x}\right) e_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_z|_{0,j,k+\frac{1}{2}}^n + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_z|_{1,j,k+\frac{1}{2}}^n \end{aligned} \quad (6.2.14c)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta y}\right) e_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta y}\right) e_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},je,k}^n + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_x|_{i+\frac{1}{2},jd,k}^n \end{aligned} \quad (6.2.14d)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta z}\right) e_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta z}\right) e_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},ke}^n + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_y|_{i,j+\frac{1}{2},kd}^n \end{aligned} \quad (6.2.14e)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta x}\right) e_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta x}\right) e_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\ = \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_z|_{ie,j,k+\frac{1}{2}}^n + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_z|_{id,j,k+\frac{1}{2}}^n \end{aligned} \quad (6.2.14f)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} e_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} + \frac{2\Delta z + v\Delta t}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,0}^n + \frac{2\Delta z - v\Delta t}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,1}^n \quad (6.2.14g)$$

$$e_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} e_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{2\Delta x + v\Delta t}{\Delta x + v\Delta t} E_y|_{0,j+\frac{1}{2},k}^n + \frac{2\Delta x - v\Delta t}{\Delta x + v\Delta t} E_y|_{1,j+\frac{1}{2},k}^n \quad (6.2.14h)$$

$$e_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} e_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{2\Delta y + v\Delta t}{\Delta y + v\Delta t} E_z|_{i,0,k+\frac{1}{2}}^n + \frac{2\Delta y - v\Delta t}{\Delta y + v\Delta t} E_z|_{i,1,k+\frac{1}{2}}^n \quad (6.2.14i)$$

$$e_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} e_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} + \frac{2\Delta z + v\Delta t}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,ke}^n + \frac{2\Delta z - v\Delta t}{\Delta z + v\Delta t} E_x|_{i+\frac{1}{2},j,kd}^n \quad (6.2.14j)$$

$$e_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} e_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{2\Delta x + v\Delta t}{\Delta x + v\Delta t} E_y|_{ie,j+\frac{1}{2},k}^n + \frac{2\Delta x - v\Delta t}{\Delta x + v\Delta t} E_y|_{id,j+\frac{1}{2},k}^n \quad (6.2.14k)$$

$$e_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} e_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{2\Delta z + v\Delta t}{\Delta z + v\Delta t} E_z|_{i,je,k+\frac{1}{2}}^n + \frac{2\Delta z - v\Delta t}{\Delta z + v\Delta t} E_z|_{i,jd,k+\frac{1}{2}}^n \quad (6.2.14l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Implicit Update

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{2\Delta z}\right) e_x|_{i+\frac{1}{2},j,0}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{2\Delta z}\right) e_x|_{i+\frac{1}{2},j,1}^{n+1} \\ = \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,0}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,1}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.15a)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{\Delta x}\right) e_y|_{0,j+\frac{1}{2},k}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{\Delta x}\right) e_y|_{1,j+\frac{1}{2},k}^{n+1} \\ = \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_y|_{0,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_y|_{1,j+\frac{1}{2},k}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.15b)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{\Delta y}\right) e_z|_{i,0,k+\frac{1}{2}}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{\Delta y}\right) e_z|_{i,1,k+\frac{1}{2}}^{n+1} \\ = \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,0,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,1,k+\frac{1}{2}}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.15c)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{\Delta z}\right) e_x|_{i+\frac{1}{2},j,ke}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{\Delta z}\right) e_x|_{i+\frac{1}{2},j,kd}^{n+1} \\ = \left(1 + \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,ke}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta z}\right) E_x|_{i+\frac{1}{2},j,kd}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.15d)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{\Delta x}\right) e_y|_{ie,j+\frac{1}{2},k}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{\Delta x}\right) e_y|_{id,j+\frac{1}{2},k}^{n+1} \\ = \left(1 + \frac{v\Delta t}{2\Delta x}\right) E_y|_{ie,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta x}\right) E_y|_{id,j+\frac{1}{2},k}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.15e)$$

$$\begin{aligned} \left(\frac{1}{2} + \frac{v\Delta t}{\Delta y}\right) e_z|_{i,je,k+\frac{1}{2}}^{n+1} + \left(\frac{1}{2} - \frac{v\Delta t}{\Delta y}\right) e_z|_{i,jd,k+\frac{1}{2}}^{n+1} \\ = \left(1 + \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,je,k+\frac{1}{2}}^{n+\frac{1}{2}} + \left(1 - \frac{v\Delta t}{2\Delta y}\right) E_z|_{i,jd,k+\frac{1}{2}}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.15f)$$

b) Explicit Update

$$e_x|_{i+\frac{1}{2},0,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} e_x|_{i+\frac{1}{2},1,k}^{n+1} + \frac{2\Delta y + v\Delta t}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},0,k}^{n+\frac{1}{2}} + \frac{2\Delta y - v\Delta t}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},1,k}^{n+\frac{1}{2}} \quad (6.2.15g)$$

$$e_y|_{i,j+\frac{1}{2},0}^{n+1} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} e_y|_{i,j+\frac{1}{2},1}^{n+1} + \frac{2\Delta z + v\Delta t}{\Delta z + v\Delta t} E_y|_{i,j+\frac{1}{2},0}^{n+\frac{1}{2}} + \frac{2\Delta z - v\Delta t}{\Delta z + v\Delta t} E_y|_{i,j+\frac{1}{2},1}^{n+\frac{1}{2}} \quad (6.2.15h)$$

$$e_z|_{0,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} e_z|_{1,j,k+\frac{1}{2}}^{n+1} + \frac{2\Delta x + v\Delta t}{\Delta x + v\Delta t} E_z|_{0,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{2\Delta x - v\Delta t}{\Delta x + v\Delta t} E_z|_{1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.15i)$$

$$e_x|_{i+\frac{1}{2},je,k}^{n+1} = \frac{v\Delta t - \Delta y}{\Delta y + v\Delta t} e_x|_{i+\frac{1}{2},jd,k}^{n+1} + \frac{2\Delta y + v\Delta t}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},je,k}^{n+\frac{1}{2}} + \frac{2\Delta y - v\Delta t}{\Delta y + v\Delta t} E_x|_{i+\frac{1}{2},jd,k}^{n+\frac{1}{2}} \quad (6.2.15j)$$

$$e_y|_{i,j+\frac{1}{2},ke}^{n+1} = \frac{v\Delta t - \Delta z}{\Delta z + v\Delta t} e_y|_{i,j+\frac{1}{2},kd}^{n+1} + \frac{2\Delta z + v\Delta t}{\Delta z + v\Delta t} E_y|_{i,j+\frac{1}{2},ke}^{n+\frac{1}{2}} + \frac{2\Delta z - v\Delta t}{\Delta z + v\Delta t} E_y|_{i,j+\frac{1}{2},kd}^{n+\frac{1}{2}} \quad (6.2.15k)$$

$$e_z|_{ie,j,k+\frac{1}{2}}^{n+1} = \frac{v\Delta t - \Delta x}{\Delta x + v\Delta t} e_z|_{id,j,k+\frac{1}{2}}^{n+1} + \frac{2\Delta x + v\Delta t}{\Delta x + v\Delta t} E_z|_{ie,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{2\Delta x - v\Delta t}{\Delta x + v\Delta t} E_z|_{id,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \quad (6.2.15l)$$

where

$$id = ie - 1, \quad jd = je - 1, \quad kd = ke - 1.$$

Appendix F

CFS-CPML for ADI-FDTD Method

By expanding (3.6.1) with some manipulations and arrangements, the full update equations for the conventional ADI-FDTD method with CFS-CPML are as follows:

For first procedure from n to $n + \frac{1}{2}$:

$$\begin{aligned}
& -\frac{a_{1,y}a_{2,y}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{y_{j-\frac{1}{2}}}}\beta E_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} - \frac{a_{1,y}a_{2,y}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{y_{j+\frac{1}{2}}}}\beta E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} + \gamma_{y_j}E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \\
& = \frac{\chi}{\beta}E_x|_{i+\frac{1}{2},j,k}^n - \frac{a_{1,z}}{\widehat{\kappa}_{z_k}}\beta \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^n \right) \\
& - \frac{a_{1,y}a_{2,x}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{x_{i+\frac{1}{2}}}}\beta \left(E_y|_{i+1,j+\frac{1}{2},k}^n - E_y|_{i,j+\frac{1}{2},k}^n - E_y|_{i+1,j-\frac{1}{2},k}^n + E_y|_{i,j-\frac{1}{2},k}^n \right) \\
& + \frac{a_{1,y}}{\widehat{\kappa}_{y_j}}\beta \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n \right) + \frac{a_1}{\beta} \left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n \right) \\
& + \frac{a_{1,y}a_2}{\widehat{\kappa}_{y_j}}\beta \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zy}}|_{i+\frac{1}{2},j-\frac{1}{2},k}^n + \psi_{h_{zx}}|_{i+\frac{1}{2},j-\frac{1}{2},k}^n \right)
\end{aligned} \tag{6.2.16a}$$

$$\begin{aligned}
& -\frac{a_{1,z}a_{2,z}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{z_{k-\frac{1}{2}}}}\beta E_y|_{i,j+\frac{1}{2},k-1}^{n+\frac{1}{2}} - \frac{a_{1,z}a_{2,z}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{z_{k+\frac{1}{2}}}}\beta E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} + \gamma_{z_k}E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \\
& = \frac{\chi}{\beta}E_y|_{i,j+\frac{1}{2},k}^n - \frac{a_{1,x}}{\widehat{\kappa}_{x_i}}\beta \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^n \right) \\
& - \frac{a_{1,z}a_{2,y}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{y_{j+\frac{1}{2}}}}\beta \left(E_z|_{i,j+1,k+\frac{1}{2}}^n - E_z|_{i,j,k+\frac{1}{2}}^n - E_z|_{i,j+1,k-\frac{1}{2}}^n + E_z|_{i,j,k-\frac{1}{2}}^n \right) \\
& + \frac{a_{1,z}}{\widehat{\kappa}_{z_k}}\beta \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n \right) + \frac{a_1}{\beta} \left(\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^n - \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^n \right) \\
& + \frac{a_{1,z}a_2}{\widehat{\kappa}_{z_k}}\beta \left(\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - \psi_{h_{xz}}|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n + \psi_{h_{xy}}|_{i,j+\frac{1}{2},k-\frac{1}{2}}^n \right)
\end{aligned} \tag{6.2.16b}$$

$$\begin{aligned}
& -\frac{a_{1,x}a_{2,x}}{\widehat{\kappa}_{x_i}\widehat{\kappa}_{x_{i-\frac{1}{2}}}}\beta E_z|_{i-1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{1,x}a_{2,x}}{\widehat{\kappa}_{x_i}\widehat{\kappa}_{x_{i+\frac{1}{2}}}}\beta E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \gamma_{x_i}E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}}
\end{aligned}$$

$$\begin{aligned}
&= \frac{\chi}{\beta} E_z|_{i,j,k+\frac{1}{2}}^n - \frac{a_{1,y}}{\widehat{\kappa}_{y_j}\beta} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^n \right) \\
&- \frac{a_{1,x}a_{2,z}}{\widehat{\kappa}_{x_i}\widehat{\kappa}_{z_{k+\frac{1}{2}}}\beta} \left(E_x|_{i+\frac{1}{2},j,k+1}^n - E_x|_{i+\frac{1}{2},j,k}^n - E_x|_{i-\frac{1}{2},j,k+1}^n + E_x|_{i-\frac{1}{2},j,k}^n \right) \\
&+ \frac{a_{1,x}}{\widehat{\kappa}_{x_i}\beta} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n \right) + \frac{a_1}{\beta} \left(\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^n - \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^n \right) \\
&+ \frac{a_{1,x}a_2}{\widehat{\kappa}_{x_i}\beta} \left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - \psi_{h_{yx}}|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n + \psi_{h_{yz}}|_{i-\frac{1}{2},j,k+\frac{1}{2}}^n \right) \quad (6.2.16c)
\end{aligned}$$

$$\begin{aligned}
H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} &= H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(E_z|_{i,j+1,k+\frac{1}{2}}^n - E_z|_{i,j,k+\frac{1}{2}}^n \right) \\
&+ \frac{a_{2,z}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}}} \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n \right) \quad (6.2.16d)
\end{aligned}$$

$$\begin{aligned}
H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} &= H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - \frac{a_{2,z}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}}} \left(E_x|_{i+\frac{1}{2},j,k+1}^n - E_x|_{i+\frac{1}{2},j,k}^n \right) \\
&+ \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n \right) \quad (6.2.16e)
\end{aligned}$$

$$\begin{aligned}
H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} &= H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(E_y|_{i+1,j+\frac{1}{2},k}^n - E_y|_{i,j+\frac{1}{2},k}^n \right) \\
&+ \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n \right) \quad (6.2.16f)
\end{aligned}$$

with the update equations of discrete variables $\psi_e^{n+\frac{1}{2}}$ and $\psi_h^{n+\frac{1}{2}}$ as

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{y_j}}{\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.17a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{z_k}}{\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.17b)$$

$$\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{z_k} \psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^n + \frac{\check{b}_{z_k}}{\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.17c)$$

$$\psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{x_i} \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^n + \frac{\check{b}_{x_i}}{\Delta x} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.17d)$$

$$\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{x_i} \psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^n + \frac{\check{b}_{x_i}}{\Delta x} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.17e)$$

$$\psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{y_j} \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^n + \frac{\check{b}_{y_j}}{\Delta y} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.17f)$$

$$\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{\Delta z} \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.17g)$$

$$\psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_z|_{i,j+1,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.17h)$$

$$\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.17i)$$

$$\psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{\Delta z} \left(E_x|_{i+\frac{1}{2},j,k+1}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.17j)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.17k)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.17l)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

$$\begin{aligned} & -\frac{a_{1,z}a_{2,z}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{z_{k-\frac{1}{2}}}}\beta E_x|_{i+\frac{1}{2},j,k-1}^{n+1} - \frac{a_{1,z}a_{2,z}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{z_{k+\frac{1}{2}}}}\beta E_x|_{i+\frac{1}{2},j,k+1}^{n+1} + \gamma_{z_k} E_x|_{i+\frac{1}{2},j,k}^{n+1} \\ & = \frac{\chi}{\beta} E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{a_{1,y}}{\widehat{\kappa}_{y_j}}\beta \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,z}a_{2,x}}{\widehat{\kappa}_{z_k}\widehat{\kappa}_{x_{i+\frac{1}{2}}}}\beta \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i+1,j,k-\frac{1}{2}}^{n+\frac{1}{2}} + E_z|_{i,j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,z}}{\widehat{\kappa}_{z_k}}\beta \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) + \frac{a_1}{\beta} \left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,z}a_2}{\widehat{\kappa}_{z_k}}\beta \left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{yx}}|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} + \psi_{h_{yz}}|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (6.2.18a)$$

$$\begin{aligned} & -\frac{a_{1,x}a_{2,x}}{\widehat{\kappa}_{x_i}\widehat{\kappa}_{x_{i-\frac{1}{2}}}}\beta E_y|_{i-1,j+\frac{1}{2},k}^{n+1} - \frac{a_{1,x}a_{2,x}}{\widehat{\kappa}_{x_i}\widehat{\kappa}_{x_{i+\frac{1}{2}}}}\beta E_y|_{i+1,j+\frac{1}{2},k}^{n+1} + \gamma_{x_i} E_y|_{i,j+\frac{1}{2},k}^{n+1} \\ & = \frac{\chi}{\beta} E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{a_{1,z}}{\widehat{\kappa}_{z_k}}\beta \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,x}a_{2,y}}{\widehat{\kappa}_{x_i}\widehat{\kappa}_{y_{j+\frac{1}{2}}}}\beta \left(E_x|_{i+\frac{1}{2},j+1,j,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - E_x|_{i-\frac{1}{2},j+1,k}^{n+\frac{1}{2}} + E_x|_{i-\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,x}}{\widehat{\kappa}_{x_i}}\beta \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) + \frac{a_1}{\beta} \left(\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,x}a_2}{\widehat{\kappa}_{x_i}}\beta \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{h_{zy}}|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \psi_{h_{zx}}|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \end{aligned} \quad (6.2.18b)$$

$$\begin{aligned} & -\frac{a_{1,y}a_{2,y}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{y_{j-\frac{1}{2}}}}\beta E_z|_{i,j-1,k+\frac{1}{2}}^{n+1} - \frac{a_{1,y}a_{2,y}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{y_{j+\frac{1}{2}}}}\beta E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} + \gamma_{y_j} E_z|_{i,j,k+\frac{1}{2}}^{n+1} \\ & = \frac{\chi}{\beta} E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{a_{1,x}}{\widehat{\kappa}_{x_i}}\beta \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \\ & - \frac{a_{1,y}a_{2,z}}{\widehat{\kappa}_{y_j}\widehat{\kappa}_{z_{k+\frac{1}{2}}}}\beta \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - E_y|_{i,j-\frac{1}{2},k+1}^{n+\frac{1}{2}} + E_y|_{i,j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \end{aligned}$$

$$\begin{aligned}
& -\frac{a_{1,y}}{\widehat{\kappa}_{y_j}\beta} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) + \frac{a_1}{\beta} \left(\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \\
& -\frac{a_{1,y}a_2}{\widehat{\kappa}_{y_j}\beta} \left(\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{xz}}|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} + \psi_{h_{xy}}|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right)
\end{aligned} \tag{6.2.18c}$$

$$\begin{aligned}
H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} &= H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \\
&+ \frac{a_{2,z}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}}} \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - E_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right)
\end{aligned} \tag{6.2.18d}$$

$$\begin{aligned}
H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} &= H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{2,z}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}}} \left(E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \\
&+ \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - E_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right)
\end{aligned} \tag{6.2.18e}$$

$$\begin{aligned}
H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} &= H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \\
&+ \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) + a_2 \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right)
\end{aligned} \tag{6.2.18f}$$

with the update equations of discrete variables ψ_e^{n+1} and ψ_h^{n+1} as

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_j}}{\Delta y} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - H_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+1} \right) \tag{6.2.19a}$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{z_k}}{\Delta z} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - H_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+1} \right) \tag{6.2.19b}$$

$$\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+1} = \check{c}_{z_k} \psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{z_k}}{\Delta z} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} - H_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+1} \right) \tag{6.2.19c}$$

$$\psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+1} = \check{c}_{x_i} \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{x_i}}{\Delta x} \left(H_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - H_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+1} \right) \tag{6.2.19d}$$

$$\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+1} = \check{c}_{x_i} \psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{x_i}}{\Delta x} \left(H_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - H_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+1} \right) \tag{6.2.19e}$$

$$\psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+1} = \check{c}_{y_j} \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{y_j}}{\Delta y} \left(H_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} - H_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+1} \right) \tag{6.2.19f}$$

$$\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{\Delta z} \left(E_y|_{i,j+\frac{1}{2},k+1}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \tag{6.2.19g}$$

$$\psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \tag{6.2.19h}$$

$$\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_z|_{i+1,j,k+\frac{1}{2}}^{n+1} - E_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \tag{6.2.19i}$$

$$\psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{\Delta z} \left(E_x|_{i+\frac{1}{2},j,k+1}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.19j)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{\Delta y} \left(E_x|_{i+\frac{1}{2},j+1,k}^{n+1} - E_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.19k)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{\Delta x} \left(E_y|_{i+1,j+\frac{1}{2},k}^{n+1} - E_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.19l)$$

Note that we assume $\sigma^m = 0$ and omit the subscript indices for media parameters ϵ , μ and σ .

Appendix G

CFS-CPML for FADI-FDTD Method

By expanding (3.6.8), upon some manipulations and arrangements, the full update equations for the FADI-FDTD method with CFS-CPML are given as

For first procedure from n to $n + \frac{1}{2}$:

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j,k}^n = \tilde{E}_x|_{i+\frac{1}{2},j,k}^n - e_x|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}} + a_1 \left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^n - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^n \right) \quad (6.2.20a)$$

$$e_y|_{i,j+\frac{1}{2},k}^n = \tilde{E}_y|_{i,j+\frac{1}{2},k}^n - e_y|_{i,j+\frac{1}{2},k}^{n-\frac{1}{2}} + a_1 \left(\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^n - \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^n \right) \quad (6.2.20b)$$

$$e_z|_{i,j,k+\frac{1}{2}}^n = \tilde{E}_z|_{i,j,k+\frac{1}{2}}^n - e_z|_{i,j,k+\frac{1}{2}}^{n-\frac{1}{2}} + a_1 \left(\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^n - \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^n \right) \quad (6.2.20c)$$

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n = \tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n-\frac{1}{2}} + a_2 \left(\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n \right) \quad (6.2.20d)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n = \tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n-\frac{1}{2}} + a_2 \left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n \right) \quad (6.2.20e)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n = \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n-\frac{1}{2}} + a_2 \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n \right) \quad (6.2.20f)$$

b) Implicit update for $\tilde{E}$

$$\begin{aligned} & -\frac{a_{1,y}a_{2,y}}{2\hat{\kappa}_{y_j}\hat{\kappa}_{y_{j-\frac{1}{2}}}\beta}\tilde{E}_x|_{i+\frac{1}{2},j-1,k}^{n+\frac{1}{2}} - \frac{a_{1,y}a_{2,y}}{2\hat{\kappa}_{y_j}\hat{\kappa}_{y_{j+\frac{1}{2}}}\beta}\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} + \frac{\gamma_{y_j}}{2}\tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \\ & = \frac{1}{\beta}e_x|_{i+\frac{1}{2},j,k}^n + \frac{a_{1,y}}{\hat{\kappa}_{y_j}\beta} \left(h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n - h_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^n \right) \\ & -\frac{a_{1,z}a_{2,z}}{2\hat{\kappa}_{z_k}\hat{\kappa}_{z_{k-\frac{1}{2}}}\beta}\tilde{E}_y|_{i,j+\frac{1}{2},k-1}^{n+\frac{1}{2}} - \frac{a_{1,z}a_{2,z}}{2\hat{\kappa}_{z_k}\hat{\kappa}_{z_{k+\frac{1}{2}}}\beta}\tilde{E}_y|_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} + \frac{\gamma_{z_k}}{2}\tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \end{aligned} \quad (6.2.21a)$$

$$= \frac{1}{\beta} e_y |_{i,j+\frac{1}{2},k}^n + \frac{a_{1,z}}{\widehat{\kappa}_{z_k} \beta} \left(h_x |_{i,j+\frac{1}{2},k+\frac{1}{2}}^n - h_x |_{i,j+\frac{1}{2},k-\frac{1}{2}}^n \right) \quad (6.2.21b)$$

$$- \frac{a_{1,x} a_{2,x}}{2 \widehat{\kappa}_{x_i} \widehat{\kappa}_{x_{i-\frac{1}{2}}}} \beta \tilde{E}_z |_{i-1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{1,x} a_{2,x}}{2 \widehat{\kappa}_{x_i} \widehat{\kappa}_{x_{i+\frac{1}{2}}}} \beta \tilde{E}_z |_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\gamma_{x_i}}{2} \tilde{E}_z |_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \\ = \frac{1}{\beta} e_z |_{i,j,k+\frac{1}{2}}^n + \frac{a_{1,x}}{\widehat{\kappa}_{x_i} \beta} \left(h_y |_{i+\frac{1}{2},j,k+\frac{1}{2}}^n - h_y |_{i-\frac{1}{2},j,k+\frac{1}{2}}^n \right) \quad (6.2.21c)$$

c) Explicit update for $\tilde{H}$

$$\tilde{H}_x |_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = 2h_x |_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{a_{2,z}}{\widehat{\kappa}_{z_{k+\frac{1}{2}}}} \left(\tilde{E}_y |_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - \tilde{E}_y |_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.22a)$$

$$\tilde{H}_y |_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = 2h_y |_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(\tilde{E}_z |_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{E}_z |_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.22b)$$

$$\tilde{H}_z |_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = 2h_z |_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{a_{2,y}}{\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(\tilde{E}_x |_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - \tilde{E}_x |_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.22c)$$

d) Explicit update for ψ_e and ψ_h

$$\psi_{e_{xy}} |_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{y_j} \psi_{e_{xy}} |_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{y_j}}{2\Delta y} \left(\tilde{H}_z |_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \tilde{H}_z |_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.23a)$$

$$\psi_{e_{xz}} |_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \check{c}_{z_k} \psi_{e_{xz}} |_{i+\frac{1}{2},j,k}^n + \frac{\check{b}_{z_k}}{2\Delta z} \left(\tilde{H}_y |_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_y |_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.23b)$$

$$\psi_{e_{yz}} |_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{z_k} \psi_{e_{yz}} |_{i,j+\frac{1}{2},k}^n + \frac{\check{b}_{z_k}}{2\Delta z} \left(\tilde{H}_x |_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_x |_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.23c)$$

$$\psi_{e_{yx}} |_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{x_i} \psi_{e_{yx}} |_{i,j+\frac{1}{2},k}^n + \frac{\check{b}_{x_i}}{2\Delta x} \left(\tilde{H}_z |_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \tilde{H}_z |_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.23d)$$

$$\psi_{e_{zx}} |_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{x_i} \psi_{e_{zx}} |_{i,j,k+\frac{1}{2}}^n + \frac{\check{b}_{x_i}}{2\Delta x} \left(\tilde{H}_y |_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_y |_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.23e)$$

$$\psi_{e_{zy}} |_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{y_j} \psi_{e_{zy}} |_{i,j,k+\frac{1}{2}}^n + \frac{\check{b}_{y_j}}{2\Delta y} \left(\tilde{H}_x |_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{H}_x |_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.23f)$$

$$\psi_{h_{xz}} |_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{xz}} |_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{2\Delta z} \left(\tilde{E}_y |_{i,j+\frac{1}{2},k+1}^{n+\frac{1}{2}} - \tilde{E}_y |_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.23g)$$

$$\psi_{h_{xy}} |_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{xy}} |_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{2\Delta y} \left(\tilde{E}_z |_{i,j+1,k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{E}_z |_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.23h)$$

$$\psi_{h_{yx}} |_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{yx}} |_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{2\Delta x} \left(\tilde{E}_z |_{i+1,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \tilde{E}_z |_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.23i)$$

$$\psi_{h_{yz}} |_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{yz}} |_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{2\Delta z} \left(\tilde{E}_x |_{i+\frac{1}{2},j,k+1}^{n+\frac{1}{2}} - \tilde{E}_x |_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.23j)$$

$$\psi_{h_{zy}} |_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}} |_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{2\Delta y} \left(\tilde{E}_x |_{i+\frac{1}{2},j+1,k}^{n+\frac{1}{2}} - \tilde{E}_x |_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.23k)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{2\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.231)$$

For second procedure from $n + \frac{1}{2}$ to $n + 1$:

a) Auxiliary (explicit) update for e and h

$$e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - e_x|_{i+\frac{1}{2},j,k}^n + a_1 \left(\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} \right) \quad (6.2.24a)$$

$$e_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} = \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - e_y|_{i,j+\frac{1}{2},k}^n + a_1 \left(\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.24b)$$

$$e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - e_z|_{i,j,k+\frac{1}{2}}^n + a_1 \left(\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.24c)$$

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^n + a_2 \left(\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.24d)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} = \tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^n + a_2 \left(\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \quad (6.2.24e)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} = \tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^n + a_2 \left(\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \quad (6.2.24f)$$

b) Implicit update for $\tilde{E}$

$$\begin{aligned} & -\frac{a_{1,z}a_{2,z}}{2\hat{\kappa}_{z_k}\hat{\kappa}_{z_{k-\frac{1}{2}}}}\beta \tilde{E}_x|_{i+\frac{1}{2},j,k-1}^{n+1} - \frac{a_{1,z}a_{2,z}}{2\hat{\kappa}_{z_k}\hat{\kappa}_{z_{k+\frac{1}{2}}}}\beta \tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} + \frac{\gamma_{z_k}}{2} \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \\ & = \frac{1}{\beta} e_x|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - \frac{a_{1,z}}{\hat{\kappa}_{z_k}\beta} \left(h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - h_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (6.2.25a)$$

$$\begin{aligned} & -\frac{a_{1,x}a_{2,x}}{2\hat{\kappa}_{x_i}\hat{\kappa}_{x_{i-\frac{1}{2}}}}\beta \tilde{E}_y|_{i-1,j+\frac{1}{2},k}^{n+1} - \frac{a_{1,x}a_{2,x}}{2\hat{\kappa}_{x_i}\hat{\kappa}_{x_{i+\frac{1}{2}}}}\beta \tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} + \frac{\gamma_{x_i}}{2} \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \\ & = \frac{1}{\beta} e_y|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{a_{1,x}}{\hat{\kappa}_{x_i}\beta} \left(h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - h_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} \right) \end{aligned} \quad (6.2.25b)$$

$$\begin{aligned} & -\frac{a_{1,y}a_{2,y}}{2\hat{\kappa}_{y_j}\hat{\kappa}_{y_{j-\frac{1}{2}}}}\beta \tilde{E}_z|_{i,j-1,k+\frac{1}{2}}^{n+1} - \frac{a_{1,y}a_{2,y}}{2\hat{\kappa}_{y_j}\hat{\kappa}_{y_{j+\frac{1}{2}}}}\beta \tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^{n+1} + \frac{\gamma_{y_j}}{2} \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} \\ & = \frac{1}{\beta} e_z|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{1,y}}{\hat{\kappa}_{y_j}\beta} \left(h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - h_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} \right) \end{aligned} \quad (6.2.25c)$$

c) Explicit update for $\tilde{H}$

$$\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = 2h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{2,y}}{\hat{\kappa}_{y_{j+\frac{1}{2}}}} \left(\tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.26a)$$

$$\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = 2h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{a_{2,z}}{\hat{\kappa}_{z_{k+\frac{1}{2}}}} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.26b)$$

$$\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = 2h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} - \frac{a_{2,x}}{\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.26c)$$

d) Explicit update for ψ_e and ψ_h

$$\psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{y_j} \psi_{e_{xy}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_j}}{2\Delta y} \left(\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - \tilde{H}_z|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n+1} \right) \quad (6.2.27a)$$

$$\psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+1} = \check{c}_{z_k} \psi_{e_{xz}}|_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} + \frac{\check{b}_{z_k}}{2\Delta z} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - \tilde{H}_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n+1} \right) \quad (6.2.27b)$$

$$\psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+1} = \check{c}_{z_k} \psi_{e_{yz}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{z_k}}{2\Delta z} \left(\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} - \tilde{H}_x|_{i,j+\frac{1}{2},k-\frac{1}{2}}^{n+1} \right) \quad (6.2.27c)$$

$$\psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+1} = \check{c}_{x_i} \psi_{e_{yx}}|_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{x_i}}{2\Delta x} \left(\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} - \tilde{H}_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.27d)$$

$$\psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+1} = \check{c}_{x_i} \psi_{e_{zx}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{x_i}}{2\Delta x} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} - \tilde{H}_y|_{i-\frac{1}{2},j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.27e)$$

$$\psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+1} = \check{c}_{y_j} \psi_{e_{zy}}|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{y_j}}{2\Delta y} \left(\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} - \tilde{H}_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^{n+1} \right) \quad (6.2.27f)$$

$$\psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{xz}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{2\Delta z} \left(\tilde{E}_y|_{i,j+\frac{1}{2},k+1}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.27g)$$

$$\psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{xy}}|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{2\Delta y} \left(\tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^{n+1} - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.27h)$$

$$\psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{yx}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{2\Delta x} \left(\tilde{E}_z|_{i+1,j,k+\frac{1}{2}}^{n+1} - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^{n+1} \right) \quad (6.2.27i)$$

$$\psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+1} = \check{c}_{z_{k+\frac{1}{2}}} \psi_{h_{yz}}|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n+\frac{1}{2}} + \frac{\check{b}_{z_{k+\frac{1}{2}}}}{2\Delta z} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k+1}^{n+1} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.27j)$$

$$\psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{y_{j+\frac{1}{2}}} \psi_{h_{zy}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{y_{j+\frac{1}{2}}}}{2\Delta y} \left(\tilde{E}_x|_{i+\frac{1}{2},j+1,k}^{n+1} - \tilde{E}_x|_{i+\frac{1}{2},j,k}^{n+1} \right) \quad (6.2.27k)$$

$$\psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+1} = \check{c}_{x_{i+\frac{1}{2}}} \psi_{h_{zx}}|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n+\frac{1}{2}} + \frac{\check{b}_{x_{i+\frac{1}{2}}}}{2\Delta x} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^{n+1} - \tilde{E}_y|_{i,j+\frac{1}{2},k}^{n+1} \right) \quad (6.2.27l)$$

For non-zero initial $\tilde{E}^0$ and $\tilde{H}^0$, we apply the initialization as follows:

$$e_x|_{i+\frac{1}{2},j,k}^{-\frac{1}{2}} = \frac{\beta}{2} \tilde{E}_x|_{i+\frac{1}{2},j,k}^0 + \frac{a_{1,z}}{2\widehat{\kappa}_{z_k}} \left(\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^0 - \tilde{H}_y|_{i+\frac{1}{2},j,k-\frac{1}{2}}^0 \right) \quad (6.2.28a)$$

$$e_y|_{i,j+\frac{1}{2},k}^{-\frac{1}{2}} = \frac{\beta}{2} \tilde{E}_y|_{i,j+\frac{1}{2},k}^0 + \frac{a_{1,x}}{2\widehat{\kappa}_{x_i}} \left(\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^0 - \tilde{H}_z|_{i-\frac{1}{2},j+\frac{1}{2},k}^0 \right) \quad (6.2.28b)$$

$$e_z|_{i,j,k+\frac{1}{2}}^{-\frac{1}{2}} = \frac{\beta}{2} \tilde{E}_z|_{i,j,k+\frac{1}{2}}^0 + \frac{a_{1,y}}{2\widehat{\kappa}_{y_j}} \left(\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^0 - \tilde{H}_x|_{i,j-\frac{1}{2},k+\frac{1}{2}}^0 \right) \quad (6.2.28c)$$

$$h_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{-\frac{1}{2}} = \frac{1}{2}\tilde{H}_x|_{i,j+\frac{1}{2},k+\frac{1}{2}}^0 + \frac{a_{2,y}}{2\widehat{\kappa}_{y_{j+\frac{1}{2}}}} \left(\tilde{E}_z|_{i,j+1,k+\frac{1}{2}}^0 - \tilde{E}_z|_{i,j,k+\frac{1}{2}}^0 \right) \quad (6.2.28d)$$

$$h_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{-\frac{1}{2}} = \frac{1}{2}\tilde{H}_y|_{i+\frac{1}{2},j,k+\frac{1}{2}}^0 + \frac{a_{2,z}}{2\widehat{\kappa}_{z_{k+\frac{1}{2}}}} \left(\tilde{E}_x|_{i+\frac{1}{2},j,k+1}^0 - \tilde{E}_x|_{i+\frac{1}{2},j,k}^0 \right) \quad (6.2.28e)$$

$$h_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^{-\frac{1}{2}} = \frac{1}{2}\tilde{H}_z|_{i+\frac{1}{2},j+\frac{1}{2},k}^0 + \frac{a_{2,x}}{2\widehat{\kappa}_{x_{i+\frac{1}{2}}}} \left(\tilde{E}_y|_{i+1,j+\frac{1}{2},k}^0 - \tilde{E}_y|_{i,j+\frac{1}{2},k}^0 \right) \quad (6.2.28f)$$

Note that we assume $\sigma^m = 0$ and omit the subscript indices for media parameters ϵ , μ and σ .

Appendix H

Stable and efficient ADI method for chip/package thermal simulation

This paper has been submitted to the 2014 IEEE Region 10 Post Graduate Student Paper contest and won the First Prize.

Stable and Efficient ADI Method for Chip/Package Thermal Simulation

Wei Choon Tay, *Student Member, IEEE*

Abstract—This paper presents a stable and efficient alternating-direction-implicit (ADI) method for chip/package thermal simulation. The formulation takes into consideration both diffusion and convection terms of the general heat transfer equation. The potential instability of the conventional Douglas-Gunn (DG) ADI method caused by the convection terms within inhomogeneous media is first alleviated. The proposed stabilized DG-ADI method is then cast into (stabilized) Peaceman-Rachford (PR) ADI method in compact form, and further formulated into stable and efficient fundamental ADI (FADI) method with operator-free right-hand-side (RHS). This results in the simplest and most concise update equation. Stability analysis by means of analyzing the eigenvalues of reduced amplification matrix verifies the stability of the FADI method, while the potential instability of the conventional DG-ADI method for inhomogeneous media is demonstrated. Furthermore, numerical results justify the high efficiency gains achievable for the FADI method over the DG-ADI and PR-ADI methods.

Index Terms—Alternating-Direction-Implicit (ADI), finite-difference methods, transient thermal simulation, temperature, stability.

I. INTRODUCTION

While technological advancement has pushed for processors with higher efficiency and performance, it involves a significant increase in power intake and packaging densities. This causes a surge in temperature for processors due to power dissipation, leading to the slow down of transistors as a result of the degradation of carrier mobility. Furthermore, there will be longer interconnect RC delays on chip or package as the interconnect metalizations increase in resistivity with rising temperature. All these will deteriorate the performance for the otherwise high performance circuit and system, not to mention the fall in reliability as the lifespan of system is shortened by the increased temperature. To resolve these problems, it is important for chip or package level thermal simulation to efficiently analyze the thermal distribution and locate the hot spots [1]–[8]. In addition, it is valuable to know the temperature profile and hot spots not only for steady state but also for transient state.

The alternating-direction-implicit (ADI) method based on Douglas-Gunn (DG) [9] and Peaceman-Rachford (PR) [10] algorithms has been proposed in [4] for thermal simulation. The ADI method has been used in thermal simulation due to its unconditional stability feature where its time step is unrestricted by the minimum spatial step in the computation domain. However, within general inhomogeneous media, the conventional DG- and PR-ADI methods are no longer unconditionally stable if both heat diffusion and convection terms are present. Moreover, the ADI method comes at an expense of increasing complexity in its implementation. Besides having to solve the tridiagonal system of equations, there are substantial arithmetic operations involved in the right-hand-side (RHS) of the update equations, not to mention the large amount of memory indexing operations incurred. These in turn increase the programming complexity and the CPU computation time.

To mitigate the high complexity of implicit methods in electromagnetics (EM), an efficient algorithm based on the fundamental ADI (FADI) method has been developed [11]. Such algorithm is included within a family of fundamental implicit schemes, which feature similar fundamental updating structures with operator-free RHS. This results in

the simplest and most concise update equation. The word “fundamental” in the acronym “FADI” is used to aptly describe the ADI method in the most fundamental (basic) form that cannot be reduced further in the RHS. Nonetheless, the conventional DG-ADI method in thermal and ADI method in EM differ in terms of their update procedures and operators. The former is not directly reducible to its fundamental form and it remains unclear on how to extend the FADI concept from EM to thermal. Furthermore, as mentioned earlier, the conventional DG-ADI method is not unconditionally stable and prior efforts are needed to ensure its stability before any improvement on the efficiency can be made.

In this paper, we shall present a stable and efficient ADI method for chip/package thermal simulation. The formulation will take into consideration both diffusion and convection terms of the general heat transfer equation. The potential instability of the conventional DG-ADI method caused by the convection terms within inhomogeneous media is first alleviated. The proposed stabilized DG-ADI method is then cast into (stabilized) PR-ADI method in compact form, and further formulated into FADI method with operator-free RHS, resulting in the simplest and most concise update equation. The relationship among temperatures resulted from these three methods, namely, DG-ADI, PR-ADI and FADI methods will be shown and discussed. Stability analysis by means of analyzing the eigenvalues of reduced amplification matrix shall be performed to verify the stability of the FADI method, while the potential instability of the conventional DG-ADI method for inhomogeneous media shall be demonstrated. Furthermore, numerical results justify the high efficiency gains achievable for the FADI method over the DG-ADI and PR-ADI methods.

II. CONVENTIONAL DG-ADI METHOD

The temperature of a system is governed by the following partial differential equation of heat transfer equation [12]–[14]:

$$\rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r},t)}{\partial t} = \nabla \cdot [\kappa(\vec{r})\nabla T(\vec{r},t)] + g(\vec{r},t) \quad (1)$$

subjected to thermal boundary condition

$$\kappa(\vec{r})\frac{\partial T(\vec{r},t)}{\partial n_\xi} + h_\xi T(\vec{r},t) = f(\vec{r},t) \quad (2)$$

where T is the time-dependent temperature at any point, κ is the thermal conductivity (W/m-K), ρ is the density of the material (kg/m³), C_p is the specific heat capacity (J/kg-K) and g is the heat energy generation rate (W/m³). $f(\vec{r},t)$ is an arbitrary function on the boundary, $\partial/\partial n_\xi$ is the differentiation along the outward direction normal to the boundary and h_ξ is the equivalent heat transfer coefficients on the boundary ($\xi = x, y$). Note that for convection boundary condition, the function $f(\vec{r},t)$ in (2) is $f(\vec{r},t) = h_\xi T_\infty$, where T_∞ is a prespecified temperature.

Expanding (1) into

$$\begin{aligned} \rho(\vec{r})C_p(\vec{r})\frac{\partial T(\vec{r},t)}{\partial t} &= \kappa(\vec{r})\nabla^2 T(\vec{r},t) \\ &+ \nabla\kappa(\vec{r}) \cdot \nabla T(\vec{r},t) + g(\vec{r},t) \end{aligned} \quad (3)$$

one can find both diffusion $\kappa(\vec{r})\nabla^2 T(\vec{r}, t)$ and convection $\nabla\kappa(\vec{r}) \cdot \nabla T(\vec{r}, t)$ terms within general inhomogeneous media.

The explicit method has the stability constraints [15]

$$\gamma = \sigma_{i,j}\Delta t \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \leq \frac{1}{2}. \quad (4)$$

$$\Delta t \leq \frac{4\sigma_{i,j}}{(|u_{i,j}| + |v_{i,j}|)^2} \quad (5)$$

where

$$\sigma_{i,j} = \frac{\kappa_{i,j}}{\rho_{i,j}C_{p,i,j}}, \quad u_{i,j} = \frac{\partial\kappa_{i,j}}{\partial x} \quad \text{and} \quad v_{i,j} = \frac{\partial\kappa_{i,j}}{\partial y}.$$

The constraints in (4) and (5) arise from the diffusion and convection terms respectively. This restricts the time step Δt for certain preset inhomogeneities and spatial steps of Δx and Δy . Therefore, applying the explicit method becomes increasingly difficult especially when one or both of the constraints become too restrictive.

Using the conventional DG-ADI method, (3) can be solved in two procedures with the inclusion of both diffusion and convection terms as

First Procedure

$$T_{\text{DG}|i,j}^{n+\frac{1}{2}} = \frac{r_{i,j}}{2} \left[\kappa_{i,j}\partial_x^2 + (\partial_x\kappa_{i,j})\partial_x \right] (T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n) + r_{i,j} \left[\kappa_{i,j}\partial_y^2 + (\partial_y\kappa_{i,j})\partial_y \right] T_{\text{DG}|i,j}^n + T_{\text{DG}|i,j}^n + G_{i,j}^n \quad (6a)$$

Second Procedure

$$T_{\text{DG}|i,j}^{n+1} = \frac{r_{i,j}}{2} \left[\kappa_{i,j}\partial_x^2 + (\partial_x\kappa_{i,j})\partial_x \right] (T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n) + \frac{r_{i,j}}{2} \left[\kappa_{i,j}\partial_y^2 + (\partial_y\kappa_{i,j})\partial_y \right] (T_{\text{DG}|i,j}^{n+1} + T_{\text{DG}|i,j}^n) + T_{\text{DG}|i,j}^n + G_{i,j}^n \quad (6b)$$

where

$$\begin{aligned} \partial_x\kappa_{i,j} &= \frac{\kappa_{i+1,j} - \kappa_{i-1,j}}{2\Delta x}, \quad \partial_y\kappa_{i,j} = \frac{\kappa_{i,j+1} - \kappa_{i,j-1}}{2\Delta y}, \\ \partial_x^2 T_{i,j}^n &= \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{\Delta x^2}, \quad \partial_x T_{i,j}^n = \frac{T_{i+1,j}^n - T_{i-1,j}^n}{2\Delta x}, \\ \partial_y^2 T_{i,j}^n &= \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{\Delta y^2}, \quad \partial_y T_{i,j}^n = \frac{T_{i,j+1}^n - T_{i,j-1}^n}{2\Delta y}, \\ r_{i,j} &= \frac{\Delta t}{\rho_{i,j}C_{p,i,j}}, \quad G_{i,j}^n = \frac{\Delta t}{\rho_{i,j}C_{p,i,j}} g_{i,j}^n. \end{aligned}$$

Note that this method can only remove the stability constraint in (4) due to the diffusion terms. To the best knowledge of the authors, many researchers omit the convection terms in the applications of conventional DG-ADI method in (6). However, if both diffusion and convection terms are included as required in the general heat transfer equation (1), the stability constraint in (5) due to the convection terms may still cause potential instability (to be demonstrated in Section V). Similar potential instability may be present in the conventional PR-ADI method as well.

III. STABILIZED DG-ADI AND PR-ADI METHODS

To completely remove both stability constraints of (4) and (5), one may carefully finite difference (1) in a direct manner without expanding into (3). Alternatively, we introduce a spatial averaging operator $\square_\xi$ to κ in (6). The proposed DG-ADI method calls for the following update procedures:

First Procedure

$$T_{\text{DG}|i,j}^{n+\frac{1}{2}} = \frac{r_{i,j}}{2} \left[(\square_\xi\kappa_{i,j})\partial_x^2 + (\partial_x\kappa_{i,j})\partial_x \right] (T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n) + r_{i,j} \left[(\square_\xi\kappa_{i,j})\partial_y^2 + (\partial_y\kappa_{i,j})\partial_y \right] T_{\text{DG}|i,j}^n + T_{\text{DG}|i,j}^n + G_{i,j}^n \quad (7a)$$

Second Procedure

$$T_{\text{DG}|i,j}^{n+1} = \frac{r_{i,j}}{2} \left[(\square_\xi\kappa_{i,j})\partial_x^2 + (\partial_x\kappa_{i,j})\partial_x \right] (T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n) + \frac{r_{i,j}}{2} \left[(\square_\xi\kappa_{i,j})\partial_y^2 + (\partial_y\kappa_{i,j})\partial_y \right] (T_{\text{DG}|i,j}^{n+1} + T_{\text{DG}|i,j}^n) + T_{\text{DG}|i,j}^n + G_{i,j}^n \quad (7b)$$

where

$$\square_x\kappa_{i,j} = \frac{\kappa_{i+1,j} + \kappa_{i-1,j}}{2}, \quad \square_y\kappa_{i,j} = \frac{\kappa_{i,j+1} + \kappa_{i,j-1}}{2}.$$

We can rearrange (7) into compact form as

$$\left(1 - \frac{1}{2}A\right)T_{\text{DG}|i,j}^{n+\frac{1}{2}} = \left(1 + \frac{1}{2}A + B\right)T_{\text{DG}|i,j}^n + G_{i,j}^n \quad (8a)$$

$$\left(1 - \frac{1}{2}B\right)T_{\text{DG}|i,j}^{n+1} = \left(1 + \frac{1}{2}A + \frac{1}{2}B\right)T_{\text{DG}|i,j}^n + \frac{1}{2}AT_{\text{DG}|i,j}^{n+\frac{1}{2}} + G_{i,j}^n \quad (8b)$$

where

$$A = r_{i,j} \left[(\square_x\kappa_{i,j})\partial_x^2 + (\partial_x\kappa_{i,j})\partial_x \right], \\ B = r_{i,j} \left[(\square_y\kappa_{i,j})\partial_y^2 + (\partial_y\kappa_{i,j})\partial_y \right].$$

Applying central approximation and upon some manipulations, we obtain

First Procedure

$$\begin{aligned} &-\frac{1}{2}\alpha_{x,i,j}T_{\text{DG}|i-1,j}^{n+\frac{1}{2}} + (1 + a_{x,i,j})T_{\text{DG}|i,j}^{n+\frac{1}{2}} - \frac{1}{2}\beta_{x,i,j}T_{\text{DG}|i+1,j}^{n+\frac{1}{2}} \\ &= \frac{1}{2}\alpha_{x,i,j}T_{\text{DG}|i-1,j}^n + \frac{1}{2}\beta_{x,i,j}T_{\text{DG}|i+1,j}^n \\ &+ \alpha_{y,i,j}T_{\text{DG}|i,j-1}^n + \beta_{y,i,j}T_{\text{DG}|i,j+1}^n \\ &+ (1 - a_{x,i,j} - 2a_{y,i,j})T_{\text{DG}|i,j}^n + G_{i,j}^n \end{aligned} \quad (9a)$$

Second Procedure

$$\begin{aligned} &-\frac{1}{2}\alpha_{y,i,j}T_{\text{DG}|i,j-1}^{n+1} + (1 + a_{y,i,j})T_{\text{DG}|i,j}^{n+1} - \frac{1}{2}\beta_{y,i,j}T_{\text{DG}|i,j+1}^{n+1} \\ &= \frac{1}{2}\alpha_{x,i,j} \left(T_{\text{DG}|i-1,j}^{n+\frac{1}{2}} + T_{\text{DG}|i-1,j}^n \right) \\ &+ \frac{1}{2}\beta_{x,i,j} \left(T_{\text{DG}|i+1,j}^{n+\frac{1}{2}} + T_{\text{DG}|i+1,j}^n \right) \\ &+ \frac{1}{2}\alpha_{y,i,j}T_{\text{DG}|i,j-1}^n + \frac{1}{2}\beta_{x,i,j}T_{\text{DG}|i,j+1}^n \\ &+ (1 - a_{x,i,j} - a_{y,i,j})T_{\text{DG}|i,j}^n - a_{x,i,j}T_{\text{DG}|i,j}^{n+\frac{1}{2}} + G_{i,j}^n \end{aligned} \quad (9b)$$

where

$$\begin{aligned} \alpha_{\xi,i,j} &= a_{\xi,i,j} - b_{\xi,i,j}, \quad \beta_{\xi,i,j} = a_{\xi,i,j} + b_{\xi,i,j}, \\ a_{\xi,i,j} &= r_{i,j} \frac{\square_\xi\kappa_{i,j}}{\Delta\xi^2}, \quad b_{\xi,i,j} = r_{i,j} \frac{\partial\xi\kappa_{i,j}}{2\Delta\xi}. \end{aligned}$$

The proposed DG-ADI method is now stabilized for general heat transfer equation including both diffusion and convection terms. The stability analysis of our proposed DG-ADI method will be presented and discussed in Section V.

Upon attaining a stabilized DG-ADI method, we find that the RHS of (9) has a substantial amount of arithmetic operations and is rather complicated. To reduce the amount of arithmetic operations and its complexity, we modify (8) into

$$\left(1 - \frac{1}{2}A\right)T_{\text{DG}|i,j}^{n+\frac{1}{2}} = 2\left(1 + \frac{1}{2}B\right)T_{\text{DG}|i,j}^n - \left(1 - \frac{1}{2}A\right)T_{\text{DG}|i,j}^n + G_{i,j}^n \quad (10a)$$

$$\left(1 - \frac{1}{2}B\right)T_{\text{DG}|i,j}^{n+1} = \frac{1}{2}\left(1 + \frac{1}{2}A + B\right)T_{\text{DG}|i,j}^n + \frac{1}{2}AT_{\text{DG}|i,j}^{n+\frac{1}{2}} + \frac{1}{2}\left(1 + \frac{1}{2}A\right)T_{\text{DG}|i,j}^n + G_{i,j}^n. \quad (10b)$$

By substituting (8a) into (10b) and upon some manipulations, we get

$$\frac{1}{2}\left(1 - \frac{1}{2}A\right)(T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n) = \left(1 + \frac{1}{2}B\right)T_{\text{DG}|i,j}^n + \frac{1}{2}G_{i,j}^n \quad (11a)$$

$$\left(1 - \frac{1}{2}B\right)T_{\text{DG}|i,j}^{n+1} = \frac{1}{2}\left(1 + \frac{1}{2}A\right)(T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n) + \frac{1}{2}G_{i,j}^n. \quad (11b)$$

Through redefinition of variables in terms of T_{PR} 's, we can cast DG-ADI method into PR-ADI method [10] in compact form as shown below:

$$\left(1 - \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} = \left(1 + \frac{1}{2}B\right)T_{\text{PR}|i,j}^n + \frac{1}{2}G_{i,j}^n \quad (12a)$$

$$\left(1 - \frac{1}{2}B\right)T_{\text{PR}|i,j}^{n+1} = \left(1 + \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} + \frac{1}{2}G_{i,j}^n. \quad (12b)$$

The proposed PR-ADI method is also stabilized for general heat transfer equation involving both diffusion and convection terms. Here, we note that the temperatures resulted from both DG-ADI and PR-ADI methods are the same at T^n and T^{n+1} , i.e.

$$T_{\text{PR}|i,j}^n = T_{\text{DG}|i,j}^n \quad (13a)$$

$$T_{\text{PR}|i,j}^{n+1} = T_{\text{DG}|i,j}^{n+1}. \quad (13b)$$

However, confusion may arise as the intermediate values $T^{n+\frac{1}{2}}$'s for both DG-ADI and PR-ADI methods are different. In fact, they have the relation

$$T_{\text{PR}|i,j}^{n+\frac{1}{2}} = \frac{1}{2}(T_{\text{DG}|i,j}^{n+\frac{1}{2}} + T_{\text{DG}|i,j}^n). \quad (13c)$$

Finally, applying central approximation and upon some manipulations for both procedures of PR-ADI method, we have

First Procedure

$$\begin{aligned} & -\frac{1}{2}\alpha_{x,i,j}T_{\text{PR}|i-1,j}^{n+\frac{1}{2}} + (1 + a_{x,i,j})T_{\text{PR}|i,j}^{n+\frac{1}{2}} - \frac{1}{2}\beta_{x,i,j}T_{\text{PR}|i+1,j}^{n+\frac{1}{2}} \\ & = \frac{1}{2}\alpha_{y,i,j}T_{\text{PR}|i,j-1}^n + \frac{1}{2}\beta_{y,i,j}T_{\text{PR}|i,j+1}^n \\ & + (1 - a_{y,i,j})T_{\text{PR}|i,j}^n + \frac{1}{2}G_{i,j}^n \end{aligned} \quad (14a)$$

Second Procedure

$$\begin{aligned} & -\frac{1}{2}\alpha_{x,i,j}T_{\text{PR}|i-1,j}^{n+1} + (1 + a_{x,i,j})T_{\text{PR}|i,j}^{n+1} - \frac{1}{2}\beta_{x,i,j}T_{\text{PR}|i+1,j}^{n+1} \\ & = \frac{1}{2}\alpha_{x,i,j}T_{\text{PR}|i-1,j}^{n+\frac{1}{2}} + \frac{1}{2}\beta_{x,i,j}T_{\text{PR}|i+1,j}^{n+\frac{1}{2}} \\ & + (1 - a_{x,i,j})T_{\text{PR}|i,j}^{n+\frac{1}{2}} + \frac{1}{2}G_{i,j}^n. \end{aligned} \quad (14b)$$

IV. STABLE AND EFFICIENT FUNDAMENTAL ADI (FADI) METHOD

Comparing (9) and (14), it can be seen that the RHS of (14) is less complicated. However, it still has considerable number of arithmetic operations. This is due to the operators that are found on the RHS of (12). To maximize efficiency, we now rewrite (12) as

$$\tilde{V}_{i,j}^n = \left(1 + \frac{1}{2}B\right)T_{\text{PR}|i,j}^n \quad (15a)$$

$$\left(1 - \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} = \tilde{V}_{i,j}^n + \frac{1}{2}G_{i,j}^n \quad (15b)$$

$$\tilde{V}_{i,j}^{n+\frac{1}{2}} = \left(1 + \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} + \frac{1}{2}G_{i,j}^n \quad (15c)$$

$$\left(1 - \frac{1}{2}B\right)T_{\text{PR}|i,j}^{n+1} = \tilde{V}_{i,j}^{n+\frac{1}{2}} \quad (15d)$$

where $\tilde{V}$'s serve as temporary auxiliary variables.

Next, we exploit the auxiliary variables in order to turn the algorithm above into a simpler one. In particular, based on (15d) at one time step backward

$$\tilde{V}_{i,j}^{n-\frac{1}{2}} = \left(1 - \frac{1}{2}B\right)T_{\text{PR}|i,j}^n \quad (16)$$

it follows that $\tilde{V}_{i,j}^n$ of (15a) is reducible to

$$\begin{aligned} \tilde{V}_{i,j}^n & = \left(1 + \frac{1}{2}B\right)T_{\text{PR}|i,j}^n \\ & = 2T_{\text{PR}|i,j}^n - \left(1 - \frac{1}{2}B\right)T_{\text{PR}|i,j}^n \\ & = 2T_{\text{PR}|i,j}^n - \tilde{V}_{i,j}^{n-\frac{1}{2}}. \end{aligned} \quad (17)$$

Furthermore, upon recognizing (15b), $\tilde{V}_{i,j}^{n+\frac{1}{2}}$ of (15c) is also reducible to

$$\begin{aligned} \tilde{V}_{i,j}^{n+\frac{1}{2}} & = \left(1 + \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} + \frac{1}{2}G_{i,j}^n \\ & = 2T_{\text{PR}|i,j}^{n+\frac{1}{2}} - \left(1 - \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} + \frac{1}{2}G_{i,j}^n \\ & = 2T_{\text{PR}|i,j}^{n+\frac{1}{2}} - \tilde{V}_{i,j}^{n+\frac{1}{2}}. \end{aligned} \quad (18)$$

Note that G is no longer required in (18), thus the FADI method requires only single heat generation input in the first procedure. With (17) and (18), algorithm (15) becomes

$$\tilde{V}_{i,j}^n = 2T_{\text{PR}|i,j}^n - \tilde{V}_{i,j}^{n-\frac{1}{2}} \quad (19a)$$

$$\left(1 - \frac{1}{2}A\right)T_{\text{PR}|i,j}^{n+\frac{1}{2}} = \tilde{V}_{i,j}^n + \frac{1}{2}G_{i,j}^n \quad (19b)$$

$$\tilde{V}_{i,j}^{n+\frac{1}{2}} = 2T_{\text{PR}|i,j}^{n+\frac{1}{2}} - \tilde{V}_{i,j}^n \quad (19c)$$

$$\left(1 - \frac{1}{2}B\right)T_{\text{PR}|i,j}^{n+1} = \tilde{V}_{i,j}^{n+\frac{1}{2}}. \quad (19d)$$

Through a simple re-definition of field variables

$$V_{i,j}^{n-\frac{1}{2}} = \frac{1}{2}\tilde{V}_{i,j}^{n-\frac{1}{2}}, \quad V_{i,j}^n = \frac{1}{2}\tilde{V}_{i,j}^n, \quad V_{i,j}^{n+\frac{1}{2}} = \frac{1}{2}\tilde{V}_{i,j}^{n+\frac{1}{2}},$$

we obtain the final update procedures as

$$V_{i,j}^n = T_{\text{F}|i,j}^n - V_{i,j}^{n-\frac{1}{2}} \quad (20a)$$

$$\left(\frac{1}{2} - \frac{1}{4}A\right)T_{\text{F}|i,j}^{n+\frac{1}{2}} = V_{i,j}^n + \frac{1}{4}G_{i,j}^n \quad (20b)$$

$$V_{i,j}^{n+\frac{1}{2}} = T_{\text{F}|i,j}^{n+\frac{1}{2}} - V_{i,j}^n \quad (20c)$$

$$\left(\frac{1}{2} - \frac{1}{4}B\right)T_{\text{F}|i,j}^{n+1} = V_{i,j}^{n+\frac{1}{2}} \quad (20d)$$

with initialization

$$V_{i,j}^{-\frac{1}{2}} = \left(\frac{1}{2} - \frac{1}{4}B\right)T_{\text{F}|i,j}^0. \quad (21)$$

We shall refer (20) as fundamental ADI, or in short, FADI method to aptly describe such ADI method in the most fundamental (basic) and simplest form that cannot be reduced further in the RHS. This is because there is no more (spatial) operator A or B to be omitted or simplified in the RHS, which results in most conciseness, efficiency and programming ease. This is unlike the DR-ADI and PR-ADI methods whose RHS still contain operators A and B pending for further reduction.

The temperatures resulted from both FADI and PR-ADI methods are the same, i.e. they have the relation

$$T_{\text{F}|i,j}^n = T_{\text{PR}|i,j}^n \quad (22a)$$

$$T_{\text{F}|i,j}^{n+\frac{1}{2}} = T_{\text{PR}|i,j}^{n+\frac{1}{2}} \quad (22b)$$

$$T_{\text{F}|i,j}^{n+1} = T_{\text{PR}|i,j}^{n+1}. \quad (22c)$$

Furthermore, the temperature acquired from the FADI method can be related to that of the DG-ADI method by

$$T_{\text{DG}|i,j}^n = T_{\text{F}|i,j}^n \quad (23a)$$

$$T_{\text{DG}|i,j}^{n+\frac{1}{2}} = 2T_{\text{F}|i,j}^{n+\frac{1}{2}} - T_{\text{F}|i,j}^n \quad (23b)$$

$$T_{\text{DG}|i,j}^{n+1} = T_{\text{F}|i,j}^{n+1}. \quad (23c)$$

Applying central approximation and upon some manipulations for (20), we have

First Procedure

$$V_{i,j}^n = T_{F|i,j}^n - V_{i,j}^{n-\frac{1}{2}} \quad (24a)$$

$$\begin{aligned} -\frac{1}{4}\alpha_{x,i,j}T_{F|i-1,j}^{n+\frac{1}{2}} + \frac{1}{2}(1+a_{x,i,j})T_{F|i,j}^{n+\frac{1}{2}} - \frac{1}{4}\beta_{x,i,j}T_{F|i+1,j}^{n+\frac{1}{2}} \\ = V_{i,j}^n + G_{i,j}^n \end{aligned} \quad (24b)$$

Second Procedure

$$V_{i,j}^{n+\frac{1}{2}} = T_{F|i,j}^{n+\frac{1}{2}} - V_{i,j}^n \quad (24c)$$

$$\begin{aligned} -\frac{1}{4}\alpha_{y,i,j}T_{F|i,j-1}^{n+1} + \frac{1}{2}(1+a_{y,i,j})T_{F|i,j}^{n+1} - \frac{1}{4}\beta_{y,i,j}T_{F|i,j+1}^{n+1} \\ = V_{i,j}^{n+\frac{1}{2}}. \end{aligned} \quad (24d)$$

For non-zero initial $T_{F|i,j}^0$, we need to apply the initialization as follows:

$$\begin{aligned} V_{i,j}^{-\frac{1}{2}} = -\frac{1}{4}\alpha_{y,i,j}T_{F|i,j-1}^0 + \frac{1}{2}(1+a_{y,i,j})T_{F|i,j}^0 \\ - \frac{1}{4}\beta_{y,i,j}T_{F|i,j+1}^0. \end{aligned} \quad (25)$$

Comparing (9), (14) and (24), we find that the last update equation (24) is obviously the most concise and simplest. Furthermore, there is a considerable decrease in the floating-point arithmetic operations. Since the overall flops count decreases, the overall efficiency increases.

V. STABILITY ANALYSIS, EFFICIENCY AND CHIP/PACKAGE THERMAL SIMULATION

A. Stability Analysis

We first demonstrate the potential instability of the conventional DG-ADI method in Section II by considering an analytical example from [16] whereby the solution to the heat transfer equation of (1) subjected to thermal boundary condition of (2) is predetermined as

$$T = \sin(2\pi v_t t) \cdot \sin(2\pi v_x x) \cdot \sin(2\pi v_y y) \quad (26)$$

where

$$v_t = 1, \quad v_x = 4, \quad v_y = 3.$$

The domain is bounded by $0 \leq x \leq 1$ m and $0 \leq y \leq 1$ m. In this paper, κ is chosen as a continuous function given by

$$\kappa(x, y) = 1.01 + \cos(30\pi x) \cdot \cos(20\pi y)$$

while constants $C_p = 1$, $\rho = 1$ and $h_\xi = 0$. Sources g and f are determined such that (1) and (2) are satisfied. The above solution is reproduced numerically using the conventional DG-ADI and FADI methods. The domain is discretized with spatial step $\Delta x = \Delta y = 0.02$ m and time step Δt is chosen as $\gamma = 100$. Figure 1 plots the transient temperature at observation point ($i = 19, j = 21$), computed by (a) conventional DG-ADI method [c.f. (6)] and (b) FADI method [c.f. (24)]. The analytical solution given in (26) is also plotted in Figure 1(b) as reference. It is observed that for conventional DG-ADI method, the computed solution has grown unbounded over time, exhibiting instability. On the other hand, the solution computed by our FADI method remains stable and is in good agreement with the analytical solution.

We then perform an independent test by finding the eigenvalues of the reduced amplification matrix. The reduced area is bounded by $0.28 \text{ m} \leq x \leq 0.52 \text{ m}$ and $0.28 \text{ m} \leq y \leq 0.52 \text{ m}$, giving the reduced amplification matrix size of $13^2 \times 13^2 = 169 \times 169$. Figure 2 now shows the scatter plot of eigenvalues of the reduced amplification matrix for (a) conventional DG-ADI method [c.f. (6)] and (b) FADI method [c.f. (24)]. It is clear that the conventional DG-ADI method again exhibits

Fig. 1. Transient temperature at observation point ($i = 19, j = 21$), computed by (a) conventional DG-ADI method [c.f. (6)] (unstable) and (b) FADI method [c.f. (24)] (stable). The FADI method is in good agreement with the analytical solution.

TABLE I
EFFICIENCY GAINS OF FADI OVER PR-ADI AND DG-ADI FOR VARIOUS COMPUTATION DOMAINS ($\gamma = 5$, 3000 TIME STEPS)

Domain Size	Efficiency Gain	
	FADI vs PR-ADI	FADI vs DG-ADI
500 × 500	1.438	2.344
1000 × 1000	1.339	2.258
1500 × 1500	1.422	2.505
2000 × 2000	1.422	2.592
2500 × 2500	1.696	3.880
3000 × 3000	2.154	6.179
3500 × 3500	3.053	7.145
4000 × 4000	4.339	7.687

instability, evident from its eigenvalues located outside the unit semi circle, whereas the eigenvalues of FADI method are still bounded within the unit semi circle. These results have further validated the stability and accuracy of our FADI method.

B. Efficiency

To justify the high efficiency gains achievable for the FADI method, we conduct numerical experiments and obtain the CPU run time of the DG-ADI, PR-ADI and FADI methods for a range of computation domains from 500×500 to 4000×4000 grids. The numerical simulation is performed for 3000 time steps with $\gamma = 5$. The programs have been compiled using Microsoft Visual C++ under Microsoft Windows 7 operating system (OS) running on Intel Dual Core 2.66 GHz processor platform.

Table I shows the CPU efficiency gains of the FADI method over the PR-ADI and DG-ADI methods for various computation domains. In particular, the efficiency gain of the FADI method over the PR-ADI method increases from 1.4 to 4.3 as the domain increases. If the FADI method is further compared to the DG-ADI method, the efficiency gain actually increases from 2.3 to 7.7! It can be seen that the DG-ADI method is the slowest among all three methods, due to its complicated

Fig. 2. Scatter plot of eigenvalues of the reduced amplification matrix for (a) conventional DG-ADI method [c.f. (6)] and (b) FADI method [c.f. (24)]. Some eigenvalues of conventional DG-ADI method are located outside unit semi circle while all eigenvalues of FADI method are located within unit semi circle.

Fig. 3. Temperature profile of a digital signal processor chip at $t=7.8\text{ms}$.

RHS which involves large amount of arithmetic operations, c.f. (9). The PR-ADI method is faster compared to the DG-ADI method as its RHS is less complicated, which in turn requires less arithmetic operations, c.f. (14). The FADI method is the fastest among all methods. As the method has operator-free RHS, it results in the simplest and most concise update equation that requires minimum number of arithmetic operations, c.f. (24).

C. Chip/Package Thermal Simulation

We now simulate a digital signal processor chip using our FADI method and the resulting temperature profile is shown in Figure 3. We consider a domain of 4000×4080 grids. The domain size is $2.58 \text{ mm} \times 2.11 \text{ mm}$ and the spatial steps are $\Delta x = 6.5 \times 10^{-4} \text{ mm}$ and $\Delta y = 5.2 \times 10^{-4} \text{ mm}$. To illustrate the nonuniform grids, several wires

connecting various functional blocks are set to have smaller spatial steps of $\Delta x = 6.5 \times 10^{-5} \text{ mm}$ and $\Delta y = 5.2 \times 10^{-5} \text{ mm}$.

The numerical simulation is also performed using both PR-ADI and DG-ADI methods and the CPU time is acquired. The efficiency gains of the FADI method over the PR-ADI and DG-ADI methods are approximately 4.6 and 8.0 times respectively. These are consistent with the CPU efficiency gains as tabulated in Table I. Exploiting the FADI method's unconditionally stable feature, we can use a high γ value ($\gamma = 50$). This makes the FADI computation much faster (44 times) than the explicit method.

VI. CONCLUSION

This paper has presented a stable and efficient ADI method for chip/package thermal simulation. The formulation has taken into consideration both diffusion and convection terms of the general heat transfer equation. The potential instability of the conventional DG-ADI method caused by the convection terms within inhomogeneous media is first alleviated. The proposed stabilized DG-ADI method is then cast into (stabilized) PR-ADI method in compact form, and further formulated into stable and efficient FADI method with operator-free RHS. This results in the simplest and most concise update equation. Stability analysis by means of analyzing the eigenvalues of reduced amplification matrix has verified the stability of the FADI method, while the potential instability of the conventional DG-ADI method for inhomogeneous media has been demonstrated. Furthermore, numerical results have justified the high efficiency gains achievable for the FADI method over the DG-ADI and PR-ADI methods.

REFERENCES

- [1] D. Chen, E. Li, E. Rosenbaum, and S. M. Kang, "Interconnect thermal modeling for accurate simulation of circuit timing and reliability," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 2, pp. 197-205, Feb. 2000.
- [2] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, "Temperature-aware microarchitecture," *Proc. of the 30th Annual Int. Symp. on Computer Architecture*, (ISCA), pp. 2-13, 2003.
- [3] T.-Y. Wang and C. C. -P. Chen, "3-D Thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1434-1445, Dec. 2002.
- [4] T.-Y. Wang and C. C. -P. Chen, "Thermal-ADI - A linear-time chip-level dynamic thermal-simulation algorithm based on alternating-direction-implicit (ADI) method," *IEEE Trans. VLSI Syst.*, vol. 11, no. 4, pp. 691-700, Aug. 2003.
- [5] P. Li, L. T. Pileggi, M. Asheghi and R. Chandra, "IC thermal simulation and modelling via efficient multigrid-based approaches," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 9, pp. 1763-1776, Sept. 2006.
- [6] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661-1675, Sept. 2007.
- [7] P.-Y. Huang, C. -K. Lin and Y. -M. Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms," *IEEE Trans. VLSI Syst.*, vol. 17, no. 5, pp. 613-626, May. 2009.
- [8] Y. Han, I. Koren, and C. M. Krishna, "TILTS: A fast architectural-level transient thermal simulation method," *Journal of Low Power Electronics*, vol. 3, pp. 1-9, 2007.
- [9] J. Douglas, Jr. and J. E. Gunn, "A general formulation of alternating direction methods-Part I: Parabolic and hyperbolic problems," *Numer. Math.*, vol. 56, no. 6, pp. 428-453, 1964.
- [10] D. W. Peaceman and H. H. Rachford, Jr., "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Ind. Appl. Math.*, vol. 3, pp. 28-41, 1955.
- [11] E. L. Tan, "Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 1, pp. 170-177, Jan. 2008.
- [12] M. N. Ozisik, *Boundary Value Problems of Heat Conduction*. New York: Dover, 1968.
- [13] M. N. Ozisik, *Finite Difference Methods in Heat Transfer*. New York: CRC, 1994.
- [14] G. E. Myers, *Analytical Methods in Conduction Heat Transfer*. 2nd ed. Schenectady, NY: Genium, 1987.
- [15] R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*. Springer-Verlag, 1984.
- [16] J. Douglas, Jr. and S. Kim, "Improved accuracy for locally one-dimensional methods for parabolic equation," *Math. Models Meth. Appl. Sci.*, vol. 11, no. 9, pp. 1563-1579, 2001.

Author's Publications

Journal Papers

1. W. C. Tay, E. L. Tan and D. Y. Heh, "Fundamental locally one-dimensional method for 3-D thermal simulation," *IEICE Trans. Electron.*, vol. E97-C, no. 7, Jul. 2014.
2. W. C. Tay and E. L. Tan, "Pentadiagonal alternating-direction-implicit finite-difference time-domain method for two-dimensional Schrödinger equation," *Comput. Phys. Comm.*, vol. 185, no. 7, pp. 1886–1892, Jul 2014.
3. W. C. Tay, D. Y. Heh and E. L. Tan, "GPU-accelerated fundamental ADI-FDTD with complex frequency shifted convolutional perfectly matched layer," *Progress In Electromagnetics Research M*, vol. 14, pp. 177–192, 2010.
4. W. C. Tay and E. L. Tan, "Implementations of PMC and PEC boundary conditions for efficient fundamental ADI- and LOD-FDTD," *Journal of Electromagnetic Waves and Applications*, vol. 24, no. 4, pp. 565–573, Mar 2010.
5. W. C. Tay and E. L. Tan, "Mur absorbing boundary condition for efficient fundamental 3-D LOD-FDTD," *IEEE Microw. Wireless Comp. Lett.*, vol. 20, no. 2, pp. 61–63, Feb. 2010.

Conference Papers

1. W. C. Tay and E. L. Tan, "Efficient algorithm for 3-D thermal alternating-direction-implicit method," *2012 IEEE Electrical Design of Advanced Packaging and Systems Symposium*, pp. 177–180, Dec. 2012.
2. D. Y. Heh, E. L. Tan and W. C. Tay, "Some recent developments in fundamental implicit FDTD schemes," *2012 Asia-Pacific Symposium on Electromagnetic Compatibility*, pp. 153–156, May 2012.

3. W. C. Tay and E. L. Tan, "Split-field PML implementation for the efficient fundamental ADI-FDTD method," *Asia Pacific Microwave Conference 2009*, pp. 1553–1556, Dec. 2009.
4. W. C. Tay and E. L. Tan, "Implementation of Mur first order absorbing boundary condition in efficient 3-D ADI-FDTD," *2009 IEEE Int. Symp. on Antennas Propag. and USNC/URSI Nat. Radio Sci. Meeting*, pp. 1-4, June 2009.

Bibliography

- [1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media", *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302–307, May. 1966.
- [2] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Norwood, MA: Artech House, 1996.
- [3] A. Taflove and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," *IEEE Trans. Microwave Theory Tech.*, vol. 23, no. 8, pp. 623–630, Aug. 1975.
- [4] G. Sun and C. W. Trueman, "Unconditionally stable Crank-Nicolson scheme for solving two-dimensional Maxwell's equations," *Electron. Lett.*, vol. 39, no. 7, pp. 595–597, Apr. 2003.
- [5] G. Sun and C. W. Trueman, "Unconditionally-stable FDTD method based on Crank-Nicolson scheme for solving three-dimensional Maxwell equations," *Electron. Lett.*, vol. 40, no. 10, pp. 589–590, May 2004.
- [6] G. Sun and C. W. Trueman, "Efficient implementations of the Crank-Nicolson scheme for the finite-difference time-domain method," *IEEE Trans. Microwave Theory Tech.*, vol. 54, no. 5, pp. 2275–2284, May 2006.
- [7] E. L. Tan, "Efficient algorithms for Crank-Nicolson-based finite-difference time-domain methods," *IEEE Trans. Microwave Theory Tech.*, vol. 56, no. 2, pp. 408–413, Feb. 2008.
- [8] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Trans. Microwave Theory Tech.*, vol. 47, no. 10, pp. 2003–2007, Oct. 1999.
- [9] F. Zheng, Z. Chen and J. Zhang, "A finite-difference time-domain method without the courant stability conditions," *IEEE Microwave Guided Wave Lett.*, vol. 9, no. 11, pp. 441–443, Nov. 1999.
- [10] F. Zheng, Z. Chen and J. Zhang, "Toward the Development of a Three-Dimensional Unconditionally Stable Finite-Difference Time-Domain Method," *IEEE Trans. Microwave Theory Tech.*, vol. 48, no. 9, pp. 1550–1558, Sep. 2000.
- [11] 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, no. 10, pp. 1743–1748, Oct. 2000.

- [12] J. Lee and B. Fornberg, "A split step approach for the 3-D Maxwell's equations," *J. Comput. Appl. Math.*, vol. 158, no. 2, pp. 485–505, Sept. 2003.
- [13] W. Fu and E. L. Tan, "Development of split-step FDTD method with higher order spatial accuracy," *Electron. Lett.*, vol. 40, no. 20, pp. 1252–1254, Sept. 2004.
- [14] W. Fu and E. L. Tan, "Compact higher-order split-step FDTD method," *Electron. Lett.*, vol. 41, no. 20, pp. 397–399, Mar. 2005.
- [15] J. Shibayama, M. Muraki, J. Yamauchi and H. Nakano, "Efficient implicit FDTD algorithm based on locally one-dimensional scheme," *Electron. Lett.*, vol. 41, no. 19, pp. 1046–1047, Sep. 2005.
- [16] E. L. Tan, "Unconditionally Stable LOD-FDTD Method for 3-D Maxwell's Equations," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 2, pp. 85–87, Feb. 2007.
- [17] M. Pharr and R. E. Fernando, *GPUGems2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Addison-Wesley, 2005.
- [18] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone and J. C. Phillips, "GPU computing," *Proceedings of the IEEE*, vol. 96, no. 5, pp. 879–899, May 2008.
- [19] Nvidia Corporation, *NVIDIA CUDA Programming Guide*, Version 2.3, Aug. 2009.
- [20] D. B. Kirk and W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann, 2010.
- [21] B. Engquist and A. Majda, "Absorbing boundary conditions for the numerical simulation of waves," *Math. Comp.*, vol. 31, no. 139, pp. 629–651, Jul. 1977.
- [22] Z. P. Liao, H. L. Wong, B. P. Yang, and Y. F. Yuan, "A transmitting boundary for transient wave analysis," *Scientia, ser. A*, vol. 27, no. 10, pp. 1063–1076, Oct. 1984.
- [23] L. Halpern and L. N. Trefethen, "Wide-angle one-way wave equations," *J. Acoust. Soc. Am.*, pp. 1397–1404, vol. 84, no. 4, Oct. 1988.
- [24] L. N. Trefethen and L. Halpern, "Well-posedness of one-way wave equations and absorbing boundary conditions," *Math. Comp.*, vol. 47, no. 176, pp. 421–435, Oct. 1986.
- [25] R. L. Higdon, "Absorbing boundary conditions for the wave equation," *Math. Comp.*, vol. 49, no. 179, pp. 65–90, Jul. 1987.
- [26] R. L. Higdon, "Numerical absorbing boundary conditions for difference approximations to the multi-dimensional wave equation," *Math. Comp.*, vol. 47, no. 176, pp. 437–459, Oct. 1986.
- [27] C. J. Railton, E. M. Daniel, D. L. Paul, and J. P. McGreehan, "Optimized absorbing boundary conditions for the analysis of planar circuits using the finite difference time domain method," *IEEE Trans. Microw. Theory Tech.*, vol. 41, no. 2, pp. 192–203, Feb. 1993.

- [28] P. Zhao and J. Litva, "A new stable and very dispersive boundary condition for the FDTD method," *IEEE MTT-S Int. Microw. Symp. Dig.*, vol. 1, pp. 35–38, May. 1994.
- [29] O. M. Ramahi, "The complementary operators method in FDTD simulations," *Antennas Propag. Mag.*, vol. 39, no. 6, pp. 33–45, Dec. 1997.
- [30] O. M. Ramahi, "The concurrent complementary operators method for FDTD mesh truncation," *IEEE Trans. Antennas Propagat.*, vol. 46, no. 10, pp. 1475–1482, Oct. 1998.
- [31] J. Y. Zhou and W. Hong, "Construction of the absorbing boundary conditions for the FDTD method with transfer functions," *IEEE Trans. Microwave Theory Tech.*, vol. 46, no. 11, pp. 1807–1809, Nov. 1998.
- [32] Z. Shao, W. Hong and J. Zhou, "Generalized Z-domain absorbing boundary conditions for the analysis of electromagnetic problems with finite-difference time-domain method," *IEEE Trans. Microwave Theory Tech.*, vol. 51, No. 1, pp. 82–90, Jan. 2003.
- [33] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time domain electromagnetic field equations," *IEEE Trans. Electromagn. Compat.*, vol. EMC-23, pp. 377–382, Nov. 1981.
- [34] A. Rahhal-Arabi and R. Mittra, "An alternative form of the Mur second-order absorbing boundary condition," *Microwave Opt. Technol. Lett.*, vol. 9, no. 6, pp. 336–338, Aug. 1995.
- [35] J. F. Mao, "Twofold Mur's First-Order ABC in the FDTD Method," *IEEE Trans. Microwave Theory Tech.*, vol. 46, no. 3, pp. 299–301, Mar. 1998.
- [36] J.-N. Hwang and F.-C. Chen, "Numerical performances of absorbing boundary conditions for ADI-FDTD method," *IEEE Int. Symp. on Antennas Propag. and USNC/URSI Nat. Radio Sci. Meeting, Albuquerque, New Mexico*, pp. 2723–2726. Jul. 2006.
- [37] J.-N. Hwang and F.-C. Chen, "Stability analysis of the Murs absorbing boundary condition in the alternating direction implicit finite-difference method," *IET Microw. Antennas Propag.*, vol. 1, no. 3, pp. 597–601, Jun. 2007.
- [38] T. Stefanski and T. D. Drysdale, "Improved Implementation of the Mur first order absorbing boundary condition in the ADI-FDTD Method," *Microwave Opt. Technol. Lett.*, vol. 50, no. 7, pp. 1757–1761, Apr. 2008.
- [39] J. P. Berenger, "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves," *J. Comput. Phys.*, vol. 114, no. 2, pp. 185–200, Oct. 1994.
- [40] J. P. Berenger, "Perfectly matched layer for the FDTD solution of wave structure interaction Problems," *IEEE Trans. Antennas Propagat.*, vol. 44, no. 1, pp. 110–117, Jan. 1996.
- [41] J. P. Berenger, "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 127, no. 2, pp. 363–379, Sept. 1996.

- [42] W. C. Chew and W. H. Weedon, "A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates," *Microwave Opt. Technol. Lett.*, vol. 7, No. 13, pp. 599–604, Sept. 1994.
- [43] C. M. Rappaport, "Perfectly matched absorbing boundary conditions based on anisotropic lossy mapping of space," *IEEE Microwave Guided Wave Lett.*, vol. 5, no. 3, pp. 90–92, Mar. 1995.
- [44] M. Kuzuoglu and R. Mittra, "Frequency dependence of the constitutive parameters of causal perfectly matched anisotropic absorbers," *IEEE Microwave Guided Wave Lett.*, vol. 6, pp. 447–449, 1996.
- [45] G. Liu and S. D. Gedney, "Perfectly matched layer media for an unconditionally stable three-dimensional ADI-FDTD Method," *IEEE Microwave Guided Wave Lett.*, vol. 10, no. 7, pp. 261–263, July 2000.
- [46] J. Roden and S. D. Gedney, "Convolution PML (CPML): an efficient FDTD implementation of CFS-PML for arbitrary media," *Microwave Opt. Technol. Lett.*, vol. 27, no. 5, pp. 334–339, Dec. 2000.
- [47] S. D. Gedney, G. Liu, J. Alan Rodden and A. Zhu, "Perfectly matched layer media with CFS for an unconditionally Stable ADI-FDTD method," *IEEE Trans. Antennas Propagat.*, vol. 49, no. 11, 1554–1559, Nov. 2001.
- [48] S. Wang and F. L. Teixeira, "An efficient PML implementation for the ADI-FDTD method," *IEEE Microw. Wireless Comp. Lett.*, vol. 13, no. 2, pp. 72–74, Feb. 2003.
- [49] V. E. do Nascimento, B.-H. V. Borges, and F. L. Teixeira, "Split-field PML implementations for the unconditionally stable LOD-FDTD method," *IEEE Microw. Wireless Comp. Lett.*, vol. 16, no. 7, pp. 398–400, Jul. 2006.
- [50] O. Ramadan, "Unsplit field implicit PML algorithm for complex envelope dispersive LOD-FDTD simulations," *Electron. Lett.*, vol. 43, no. 5, pp. 17–18, Mar. 2007.
- [51] I. Ahmed, E. P. Li, and K. Krohne, "Convolutional perfectly matched layer for an unconditionally stable LOD-FDTD method," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 12, pp. 816–818, Dec. 2007.
- [52] M. N. Ozisik, *Boundary value problems of heat conduction*, New York: Dover, 1968.
- [53] G. E. Myers, *Analytical methods in conduction heat transfer*, 2nd ed. Schenectady, NY: Genium, 1987.
- [54] M. D. Mikhailov and M. N. Ozisik, *Unified Analysis and Solutions of Heat and Mass Diffusion*, John Wiley and Sons, 1984.
- [55] M. N. Ozisik, *Finite difference methods in heat transfer*, New York: CRC, 1994.
- [56] A. Mitchell and D. Griffiths, *The Finite Difference Method in Partial Differential Equations*, John Wiley & Sons, 1980.
- [57] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type," *Advances in Computational Mathematics*, vol. 6, no. 1, pp. 207–226, 1996.

- [58] G. W. Recktenwald, "Finite-difference approximations to the heat equation," Class Notes 2004.
- [59] N. E. Mastorakis, "An extended Crank-Nicolson method and its applications in the solution of partial differential equations: 1-D and 3-D conduction equations," *Proceedings of the 10th WSEAS International Conference on APPLIED MATHEMATICS*, pp. 134–143, Nov. 2006.
- [60] J. Douglas, Jr. and J. E. Gunn, A general formulation of alternating direction methods—Part I: Parabolic and hyperbolic problems, *Numer. Math.*, vol. 56, no. 6, pp. 428–453, 1964.
- [61] D. W. Peaceman, H. H. Rachford, Jr., "The numerical solution of parabolic and elliptic differential equations," *J. Soc. Ind. Appl. Math.*, vol. 3, no. 1, pp. 28–41, Mar. 1955.
- [62] M. J. Chang, L. C. Chow and W. S. Chang, "Improved alternating-direction implicit method for solving transient three-dimensional heat diffusion problems," *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*, vol. 19, no. 1, pp. 69–84, 1991.
- [63] T. -Y. Wang, C. C. -P. Chen, "Thermal-ADI-A linear-time chip-level dynamic thermal-simulation algorithm based on alternating-direction-implicit (ADI) method," *IEEE Trans. VLSI Syst.*, vol. 11, no. 4, pp. 691–700, Aug. 2003.
- [64] T. -Y. Wang and C. C. -P. Chen, "3-D Thermal-ADI: A linear-time chip level transient thermal simulator," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1434–1445, Dec. 2002.
- [65] S. Brandt and H. D. Dahmen, The picture book of quantum mechanics, Springer-Verlag, Berlin, 1995.
- [66] D. J. Griffiths, Introduction to quantum mechanics, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [67] D.M. Sullivan, D. S. Citrin, "Determining quantum eigenfunctions in three-dimensional nanoscale structures," *J. Appl. Phys.*, vol. 97, no. 10, pp. 1–6, May 2005.
- [68] F. Schwabl, "Advanced Quantum Mechanics," Springer-Verlag Berlin Heidelberg, 2000
- [69] G. D. Akrivis, "Finite difference discretization of the cubic schrödinger equation," *IMA Journal of Numerical Analysis*, vol. 13, no. 1, pp. 115–124, Jan. 1993.
- [70] D. F. Griffiths, A. R. Mitchell and J. L. Morris, "A numerical study of the nonlinear Schrödinger equation," *Computer Meths. App. Mech. Engrg*, vol. 45, no. 1, pp. 177–215, 1984.
- [71] J. Argyris and M. Haase, "An engineers guide to soliton phenomena: Application of the finite element method," *Computer Meths. App. Mech. Engrg*, vol. 61, no. 1, pp. 71–122, 1987.
- [72] A. B. Shamardan, "The numerical treatment of the nonlinear Schrödinger equation," *Computers & Mathematics with Applications*, vol. 19, no. 7, pp. 67–73, 1990.

- [73] A. S. Fokas, "An initial-boundary value problem for the nonlinear Schrödinger equation," *Physica D: Nonlinear Phenomena*, vol. 35, no. 1–2, pp. 167–185, Apr. 1989.
- [74] R. F. Bikbaev and V. O. Tarasov, "Initial boundary value problem for the nonlinear Schrödinger equation," *J. Phys. A: Math. Gen.*, vol. 24, no. 11, pp. 2507–2516, 1991.
- [75] R. L. Burden and J. D. Faires, *Numerical Analysis*, Brooks/Cole: Boston, 2011.
- [76] W. Guo, J. Lan, X. Wang, Y. Peng and W. Sui, "Analysis of finite difference time domain technique to solve the time-dependent Schrödinger equation in quantum structures in inhomogeneous medium," *Nanoelectronics Conference (INEC), 2010 3rd International*, pp. 3–8, Jan. 2010.
- [77] C. Leforestier, R. H. Bisseling, C. Cerjan, M. D. Feit, R. Friester, et al., "A Comparison of Different Propagation Schemes for the Time Dependent Schrödinger Equation," *J. Comput. Phys.*, vol. 94, no. 1, pp. 59–80, May. 1991.
- [78] Z. Chen, J. Zhang and Z. Yu, "Solution of the time-dependent Schrödinger equation with absorbing boundary conditions," *J. Semicond.*, vol. 30, no. 1, pp. 1–6, Jan. 2009.
- [79] P. B. Visscher, "A fast explicit algorithm for the time-dependent Schrödinger equation," *Comput. Phys.*, vol. 5, no. 6, pp. 596–598, Nov. 1991.
- [80] A. Soriano, E.A. Navarro, J.A. Portí and V. Such, "Analysis of the finite difference time domain technique to solve the Schrödinger equation for quantum devices," *J. Appl. Phys.*, vol. 95, no. 12, pp. 8011–8018, Apr. 2004.
- [81] D. M. Sullivan and D. S. Citrin, "Determining a complete three-dimensional set of eigenfunctions for nanoscale structure analysis," *J. Appl. Phys.*, vol. 98, no. 8, art. no. 084311. Oct. 2005.
- [82] Y. C. N. Na and Y. Yamamoto, "Time-domain simulation of Schrödinger equation to determine the effective potential induced by an oscillating standing wave," *J. Appl. Phys.*, vol. 100, no. 4, art. no. 044901. Aug. 2006.
- [83] Y-K. Ee, Y. P. Gupta, R. A. Arif and N. Tansu, "Quantum 3-D finite-difference-time-domain (Q-FDTD) analysis of InGaAs-GaAsP quantum dots nanostructures," *Proc. of SPIE - Int. Soc. Optic. Eng.*, vol. 6468, Jan. 2007.
- [84] D. M. Sullivan, *Electromagnetic simulation using the FDTD method*, IEEE Press, New York, 2000.
- [85] W. Dai, G. Li, R. Nassar and S. Su, "On the stability of the FDTD method for solving a time-dependent Schrödinger equation," *Numer. Methods Partial Differential Equations*, vol. 21, no. 6, pp. 1140–1154, Apr. 2005.
- [86] L. Lapidus and G. F. Pinder, *Numerical Solution of Partial Differential Equation in Science and Engineering*, Wiley: New York, 1982.
- [87] T. R. Taha and M. I. Ablowitz, "Analytical and Numerical Aspects of Certain Nonlinear Evolution Equations. II. Numerical, Nonlinear Schrödinger Equation," *J. Comput. Phys.*, vol. 55, no. 2, pp. 203–230, Aug. 1984.

- [88] Q.S. Chang, Erhui Jia and W. Sun, "Difference schemes for solving the generalized nonlinear Schrödinger equation," *J. Comput. Phys.*, vol. 148, no. 2, pp. 397–415, Jan. 1999.
- [89] A. Ashyralyev and A. Sirma, "Modified Crank-Nicolson difference schemes for nonlocal boundary value problem for the Schrödinger equation," *Discrete Dynamics in Nature and Society*, at. no. 584718, 2009.
- [90] D. Kang and E. Won, "Precise numerical solutions of potential problems using the Crank-Nicolson method," *J. Comput. Phys.*, vol. 227, no. 5, pp. 2970–2976, Feb. 2008.
- [91] M. Subaşı, "On the finite-differences schemes for the numerical solution of two dimensional Schrödinger equation," *Numer. Methods Partial Differential Equations*, vol. 18, no. 6, pp. 752–758, Nov. 2002.
- [92] M. Dehghan, "Finite difference procedures for solving a problem arising in modeling and design of certain optoelectronic devices," *Math. Comput. Simulation*, vol. 71, no. 1, pp. 16–30, Mar. 2006.
- [93] H.Q. Wang, "Numerical studies on the split-step finite difference method for nonlinear Schrödinger equations," *Appl. Math. Comput.*, vol. 170, no. 1, pp. 17–35, Nov. 2005.
- [94] M. Dehghan and A. Taleei, "A compact split-step finite difference method for solving the nonlinear Schrödinger equations with constant and variable coefficients," *Comput. Phys. Comm.*, vol. 181, no. 1, pp. 43–51, Jan. 2010.
- [95] K. Malakuti and E. Parilov, "A split-step finite difference method for non-paraxial nonlinear Schrödinger equation at critical dimension," *Applied Numer. Math.*, vol. 61, no. 7, pp. 891–899, Jul. 2011.
- [96] E. L. Tan, "Efficient Algorithm for the Unconditionally Stable 3-D ADI-FDTD Method," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 1, pp. 7–9, Jan. 2007.
- [97] E. L. Tan, "Fundamental schemes for efficient unconditionally stable implicit finite-difference time-domain methods," *IEEE Trans. Antennas Propagat.*, vol. 56, no. 1, pp. 170–177, Jan. 2008.
- [98] M. J. Inman and A. Z. Elsherbeni, "Programming video cards for computational electromagnetics applications," *Antennas Propag. Mag.*, vol. 47, no. 6, pp. 71–78, Dec 2005.
- [99] Y. B. Tao, H. Lin and H. J. Lin, "From CPU to GPU: GPU-based electromagnetic computing (GPUECO)," *Progress In Electromagnetics Research*, PIER 81, pp. 1–19, 2008.
- [100] P. Sypek, A. Dziekonski and M. Mrozowski, "How to render FDTD computations more effective using a graphics accelerator," *IEEE Trans. Magnetics*, vol. 45, no. 3, pp. 1324–1327, Mar. 2009.
- [101] Y. Zhang, J. von Hagen, M. Younis, C. Fischer and W. Wiesbeck, "Planar Artificial Magnetic Conductors and Patch Antennas," *IEEE Trans. Antennas Propagat.*, vol. 51, no. 10, pp. 2704–2712, Oct. 2003.

- [102] I. V. Lindell and A. H. Sihvola, "Perfect electromagnetic conductor," *J. Electrom. Waves Applicat.*, vol. 19, no. 7, pp. 861–869, 2005.
- [103] A. Hussain and Q. A. Naqvi, "Perfect electromagnetic conductor (PEMC) and fractional waveguide," *Progress In Electromagnetics Research*, PIER 73, pp. 61–69, 2007.
- [104] W. Yu, R. Mittra, T. Su, Y. Liu and X. Yang, *Parallel Finite-Difference Time-Domain Method*, 1st ed. Artech House Publishers, 2006.
- [105] J.D. Hoffman, Numerical Methods for Engineers and Scientists, New York: McGraw-Hill, 1992.
- [106] F. Liang and G. Wang, "Study of Mur's and UPML absorbing boundary condition for the LOD-FDTD Method," in *Proc. Int. Conf. Microw. Technol.*, vol. 2, pp. 947-949, Apr. 2008.
- [107] E. Isaacson, Analysis of Numerical Methods, Dover Publication, 1994.
- [108] G. D'Inzeo, F. Giannini and R. Sorrentino, "Novel microwave integrated low-pass filters," *Electron. Lett.*, vol. 15, no. 9, pp. 258–260, Apr. 1979.
- [109] D. M. Sheen, S. M. Ali, M. D. Abouzahra and J. A. Kong, "Application of the three-dimensional finite-difference time-domain method to the analysis of planar microstrip circuits," *IEEE Trans. Microwave Theory Tech.*, vol. 38, no. 7, pp. 849–857, Jul. 1990.
- [110] V. Szekely, "Tracing the thermal behavior of ICs," *IEEE Design Test Comput.*, vol. 15, no. 2, pp. 14–21, Apr.-Jun. 1998.
- [111] K. Banerjee, A. Mehrotra, A. Sangiovanni-Vincentelli, and C. Hu, "On thermal effects in deep sub-micron vlsi interconnects," *36th ACM/IEEE Design Automation Conf.*, pp. 885891, Jun. 1999.
- [112] K. Banerjee, M. Pedram, and A. H. Ajami, "Analysis and optimization of thermal issues in high-performance vlsi," *ACM/SIGDA Int. Symp. Physical Design (ISPD)*, pp. 230–237, Apr. 2001.
- [113] K. Banerjee and A. Mehrotra, "Global (interconnect) warming," *IEEE Circuits Devices Mag.*, vol. 17, no. 5, pp. 16–32, Sept. 2001.
- [114] G. Huang, M. Bakir, A. Naeemi, H. Chen and J.D. Meindl, "Power delivery for 3D chip stacks: Physical modeling and design implications," *Proc. IEEE Electrical Performance of Electronic Packaging*, IEEE CS Press, pp. 205–208, Oct. 2007.
- [115] B. Black ., M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCauley, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, and C. Webb, "Die stacking (3D) Microarchitecture," *Proc. 39th Int'l Symp. Microarchitecture*, ACM Press, pp. 469–479, Dec. 2006.
- [116] W. R. Davis, E. C. Oh, A. M. Sule and P. D. Franzon, "Application exploration for 3-D integrated circuits: TCAM, FIFO, and FFT case studies," *IEEE Trans. VLSI Syst.*, vol. 17, no. 4, pp. 496–506, Apr. 2009.

- [117] K. Puttaswamy and G. H. Loh, "3D-Integrated SRAM components for high-performance microprocessors," *IEEE Trans. on Computers*, vol. 58, no. 10, pp. 1369-1381, Oct. 2009.
- [118] J. L. Ayala, A. Sridhar and D. Cuesta, "Thermal modeling and analysis of 3D multi-processor chip," *Integration, the VLSI Journal*, vol. 43, no. 4, pp. 327-341, Sept. 2010.
- [119] T. -M. Chen, "A modified hybrid Laplace transform finite element method for transient heat conduction problems," *Comput. Meth. Appl. Mech. Engrg.*, vol. 98, no. 2, pp. 261-272, Jul. 1992.
- [120] A. Kharab, "Use of multiple sheets for the solution of a three-dimensional transient heat conduction problem," *Comput. Math. Appl.*, vol. 34, no. 1, pp. 71-79, Jul. 1997.
- [121] D. Chen, E. Li, E. Rosenbaum and S. M. Kang, "Interconnect thermal modeling for accurate simulation of circuit timing and reliability," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 19, no. 2, pp. 197-205, Feb. 2000.
- [122] F. Ilincai and J.-F. Hétu, "Galerkin gradient least-squares formulations for transient conduction heat transfer," *Comput. Meth. Appl. Mech. Engrg.*, vol. 191, no. 27-28, pp. 3073-3097, Apr. 2002.
- [123] P. Li, L. T. Pileggi, M. Asheghi and R. Chandra, "IC thermal simulation and modelling via efficient multigrid-based approaches," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 9, pp. 1763-1776, Sept. 2006.
- [124] Y. Han, I. Koren, and C. M. Krishna, "TILTS: A fast architectural-level transient thermal simulation method," *Journal of Low Power Electronics*, vol. 3, no. 1, pp. 1-9, 2007.
- [125] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661-1675, Sept. 2007.
- [126] A. Gorguis and W. K. Benny Chan, "Heat equation and its comparative solutions," *Comput. Math. Appl.*, vol. 55, no. 12, pp. 2973-2980, Jun. 2008.
- [127] F. S. Loureiro, W. J. Mansur and C. A. B. Vasconcellos, "A hybrid time/Laplace integration method based on numerical Greens functions in conduction heat transfer," *Comput. Meth. Appl. Mech. Engrg.*, vol. 198, no. 33-36, pp. 2662-2672, Jul. 2009.
- [128] Z. Lei, N. Pan and S. Zhu, "Transient methods of thermal properties measurement on fibrous materials," *J. of Heat Transfer*, vol. 132, no. 3, art. no. 032601, Dec. 2009.
- [129] P. -Y. Huang and Y. -M. Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms," *IEEE Trans. VLSI Syst.*, vol. 17, no. 5, pp. 613-626, May 2009.
- [130] H.-C. Chang, P.-Y. Huang, T.-J. Li, and Y.-M. Lee, "Statistical electro-thermal analysis with high compatibility of leakage power models," *Proc. IEEE Int. Syst. on Chip Conf. (SOCC)*, pp. 139-144, Sept, 2010.

- [131] W. Li, B. Yu, X. Wang, P. Wang and S. Sun, "A finite volume method for cylindrical heat conduction problems based on local analytical solution," *Int. J. Heat Mass Transfer*, vol. 55, no. 21–22, pp. 5570–5582, Oct. 2012.
- [132] F. Frantz, L. Labrak and I. O'Connor, "3D IC floorplanning: Automating optimization settings and exploring new thermal-aware management techniques," *Microelectronics Journal*, vol. 43, no. 6, pp. 423–432, Mar. 2012.
- [133] D. Cuesta, J. L. Risco-Martin and J. L. Ayala, "3D thermal-aware floorplanner using a MILP approximation," *Microprocessors and Microsystems*, vol. 36, no. 5, pp. 344–354, Feb. 2012.
- [134] N. R. D. Gracia, S. Rajaram, Nivethitha and A. Sudarsan, "Thermal aware modern VLSI floorplanning," *Int. Conf. on Devices, Circuits and Systems, ICDCS*, Coimbatore, India pp. 187–190, Mar. 2012.
- [135] A. Malek, Z. K. Bojdi and P. N. N. Golbarg, "Solving fully three-dimensional microscale dual phase lag problem using mixed-collocation, finite difference discretization," *J. of Heat Transfer*, vol. 134, no. 9, art. no. 094504, Jul. 2012.
- [136] Z. Feng and P. Li, "Fast Thermal Analysis on GPU for 3D ICs With Integrated Microchannel Cooling," *IEEE Trans. VLSI Syst.*, vol. 21, no. 8, pp. 1526–1539, Aug. 2013.
- [137] X.-X. Liu, Z. Liu, X.-D. Tan and J. Gordon, "Full-chip Thermal Analysis of 3D ICs with Liquid Cooling by GPU-Accelerated GMRES Method," in *Proc. Int. Symp. Quality Electron. Design (ISQED)*, Santa Clara, CA, USA, vol. 19, no. 2, pp. 123–128, Mar. 2012.
- [138] J.-M. Koo, S. Im, L. Jiang and K. E. Goodson, "Integrated Microchannel Cooling for Three-Dimensional Electronic Circuit Architectures," *ASME J. Heat Transfer*, vol. 127, no. 1, pp. 49–58, Feb. 2005.
- [139] R. Peyret and T. D. Taylor, *Computational methods for fluid flow*, Springer-Verlag, 1984.
- [140] E. L. Tan, "Concise Current Source Implementation for Efficient 3-D ADI-FDTD Method," *IEEE Microw. Wireless Comp. Lett.*, vol. 17, no. 11, Nov. 2007.
- [141] J. Douglas, Jr. and S. Kim, "Improved accuracy for locally one-dimensional methods for parabolic equation," *Math. Models Meth. Appl. Sci.*, vol. 11, no. 9, pp. 1563–1579, 2001.
- [142] M. Kaviany, *Principles of Heat Transfer*, New York: John Wiley & Sons, 2002.
- [143] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, "Temperature-aware microarchitecture," *Procs. of the 30th Annual Int. Symp. on Computer Architecture (ISCA)*, vol. 31, no. 2, pp. 2–13, May 2003.
- [144] K. Sankaranarayanan, S. Velusamy, M.R. Stan and K. Skadron, "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," *J. Instruction-Level Parallelism*, vol. 8, no. 1, pp. 1–16, Oct. 2005.

- [145] J. Long, S.O. Memik, G. Memik, and R. Mukherjee, "Thermal monitoring mechanisms for chip multiprocessors," *Trans. on Architecture and Code Optimization*, vol. 5, no. 9, pp. 1–33, Aug. 2008.
- [146] Y.V. Kopylov, A.V. Popov and A.V. Vinogradov, "Applications of the parabolic wave equations to X-ray diffraction optics," *Optics Comm.*, vol. 118, no. 5–6, pp. 619–636, Aug. 1995.
- [147] W. Huang, C. Xu, S.T. Chu and S.K. Chaudhuri, "The finite-difference vector beam propagation method," *J. Lightwave Technol.*, vol. 10, no. 3, pp. 295–304, Mar. 1992.
- [148] G.P. Agrawal, *Nonlinear Fiber Optics*, 3rd ed., Academic Press, San Diego, 2001.
- [149] C. L. Gardner, "Quantum hydrodynamic model for semiconductor devices," *SIAM J. App. Math.*, vol. 54, no. 2, Apr. 1994.
- [150] Z. Cai, Y. Fan, R. Li, T. Lu and Y. Wang, "Quantum hydrodynamics models by moment closure of wigner equation," *J. Math. Phys.*, vol. 53, no. 10, Sep. 2012.
- [151] J. Wheatley, D.S. Buchanan, G.W. Swift, A. Migliori and T. Hofler, "Nonlinear natural engine: Model for thermodynamic processes in mesoscale systems," *Proc. Natl. Acad. Sci. USA*, vol. 82, no. 23, pp. 7805–7809, Dec. 1985.
- [152] M. Dehghan and D. Mirzaei, "The meshless local Petrov-Galerkin (MLPG) method for the generalized two-dimensional non-linear Schrödinger equation," *Eng. Anal. with Boundary Elem.*, vol. 32, no. 9, pp. 747–756, Jan. 2008.
- [153] M. Dehghan and D. Mirzaei, "Numerical solution to the unsteady two-dimensional Schrödinger equation using meshless local boundary integral equation method," *Int. J. Numer. Methods Eng.*, vol. 76, no. 4, pp. 501–520, Oct. 2008.
- [154] A. Mohebbi and M. Dehghan, "The use of compact boundary value method for the solution of two-dimensional Schrödinger equation," *J. Comput. Appl. Math.*, vol. 225, no. 1, pp. 124–134, Mar. 2009.
- [155] M. Dehghan and A. Taleei, "Numerical solution of nonlinear Schrödinger equation by using time-space pseudo-spectral method," *Numer. Methods Partial Differ. Equations*, vol. 26, no. 4, pp. 979–992, Jul. 2010.
- [156] J. Jin and X. Wu, "Analysis of finite element method for one-dimensional time-dependent Schrödinger equation on unbounded domain," *J. Comput. Appl. Math.*, vol. 220, no. 1–2, pp. 240–256, Oct. 2008.
- [157] X. Gong, L. Shen, D. Zhang and A. Zhou, "Finite element approximations for Schrödinger equations with applications to electronic structure computations," *J. Comp. Math.*, vol. 26, no. 3, pp. 310–323, Mar. 2008.
- [158] H. Chen, F. Liu and A. Zhou, "A two-scale higher-order finite element discretization for Schrödinger equation," *J. Comp. Math.*, vol. 27, no. 2, pp. 315–337, Mar. 2009.

- [159] T.E. Simos and P.S. Williams, "On finite difference methods for the solution of the Schrödinger equation," *Computers and Chemistry*, vol. 23, no. 6, pp. 513–554, Nov. 1999.
- [160] I. W. Sudiarta and D. J. W. Geldart, "Solving the Schrödinger equation using the finite difference time domain method," *J. Phys. A: Math. Theor.*, vol. 40, no. 8, pp. 1885–1896, Feb. 2007.
- [161] M. Dehghan, "Alternating direction implicit methods for two-dimensional diffusion with a non-local boundary condition," *Int. J. Comput. Math.*, vol. 72, no. 3, pp. 349–366, 1999.
- [162] M. Dehghan, "A new ADI technique for two-dimensional parabolic equation with an integral condition," *Comput. Math. Appl.*, vol. 43, no. 12, pp. 1477–1488, Jun. 2002.
- [163] Y. Xu and L. Zhang, "Alternating direction implicit method for solving two-dimensional cubic nonlinear Schrödinger equation," *Comput. Phys. Comm.*, vol. 183, no. 5, pp. 1082–1093, May 2012.
- [164] M. Dehghan and A. Shokri, "A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions," *Comput. Math. Appl.*, vol. 54, no. 1, pp. 136–146, Jul. 2007.
- [165] M. Qiu, "Analysis of guided modes in photonic crystal fibers using the finite-difference time-domain method," *Microwave Opt. Technol. Lett.*, vol. 30, no. 5, pp. 327–330, Sep. 2001.
- [166] S. Lou, Z. Wang, G. Ren and S. Jian, "Compact 2D FDTD method for modeling photonic crystal fibers," *Proc. SPIE-Int. Soc. Optic. Engineering*, vol. 5279, pp. 158–164, Apr. 2004.
- [167] J. J. Hu, G. Ren, X. Yu, G. Wang, P. P. Shum, C. Lu, K. T. V. Grattan and T. Sun, "A generalized 2D FDTD model for photonic crystal fibers with frequency dependent media," *Opt. Quant. Electron.*, vol. 39, no. 12–13, pp. 1133–1143, Oct. 2007.
- [168] H. Rao, R. Scarmozzino and R. M. Osgood, Jr., "An improved ADI-FDTD method and its application to photonic simulations," *IEEE Photon. Tech. Lett.*, vol. 14, no. 4, pp. 477–479, Apr. 2002.
- [169] Y. Zhao and H. Yang, "Finite-difference time-domain study of guided modes in nano-plasmonic waveguides," *IEEE Trans. Antennas Propagat.*, vol. 55, no. 11, pp. 3070–3077, Nov. 2007.
- [170] I. Ahmed, E. H. Khoo, E. Li and R. Mittra, "A hybrid approach for solving coupled Maxwell and Schrödinger equations arising in the simulation of nano-devices," *IEEE Antennas. Wireless Propagat. Lett.*, myvol9, pp. 914–917, 2010
- [171] P. Turati and Y. Hao, "A FDTD solution to the Maxwell-Schrödinger coupled model at the microwave range," *Proc. of the 2012 Int'l. Conf. on Electromagn. in Adv. Applicat., ICEAA'12*, pp. 363–366, Sept. 2012.
- [172] L. Pierantoni, D. Mencarelli and T. Rozzi, "A new 3-D transmission line matrix scheme for the combined Schrödinger-Maxwell problem in the electronic/electromagnetic characterization of nanodevices," *IEEE Trans. Microwave Theory Tech.*, vol. 56, no. 3, pp. 654–662, Mar. 2008.

- [173] L. Pierantoni, D. Mencarelli and T. Rozzi, “Boundary immittance operators for the Schrödinger-Maxwell problem of carrier dynamics in nanodevices,” *IEEE Trans. Microwave Theory Tech.*, vol. 57, no. 5, pp. 1147–1155, May. 2009.
- [174] W. Sui, J. Yang, X. H. Yun and C. Wang, “Including quantum effects in electromagnetic system - an FDTD solution to Maxwell -Schrödinger equations,” *IEEE IMS*, pp. 1979–1982, Jun. 2007.