NOTE TO USERS

Development and Evaluation of Novel Finite-Difference Time-Domain Methods for Solving Maxwell's Equations

Guilin Sun

A Thesis
In
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy (Electrical Engineering) at
Concordia University, Montreal, Quebec, Canada

November 2004

Your file Votre référence
ISBN: 0-494-04059-9
Our file Notre référence
ISBN: 0-494-04059-9

ABSTRACT

Development and Evaluation of Novel Finite-Difference Time-Domain Methods for Solving Maxwell's Equations

Guilin Sun, Ph.D.

Concordia University, 2005

This thesis proposes several new finite-difference time-domain (FDTD) methods to overcome shortcomings of current FDTD schemes: the new explicit methods have better numerical accuracy and the new implicit methods have unconditional stability; an error quantification method is described to evaluate the discretization error of a FDTD method; and a new concept of numerical loss in lossy materials is discussed, which has been neglected by the FDTD community.

The new explicit methods are derived by optimizing the numerical dispersion relation. The 24-stencil method and the neighborhood-average method can have high accuracy in a given angular sector; or have zero anisotropy in the 2D and 3D cases. Combining the two methods, the neighborhood-average-24 method provides one order-of-magnitude lower accumulated phase error than other published methods, and can use as large as the Courant time step size.

The correct numerical dispersion relations for the implicit alternating-direction-implicit (ADI) method are derived and verified with good agreement with the numerical experiments. The inconsistency in the literature concerning the dispersion relation is removed.

Based on the high-accuracy, fully implicit and inefficient Crank-Nicolson scheme, several new efficient implicit methods are proposed, which have much smaller anisotropy and smaller discretization error than ADI. The numerical dispersion relations and the perturbation errors to the Crank-Nicolson scheme are given.

It is shown that all the unconditionally-stable methods have their own time-step-size upper bounds to avoid non-physical attenuation, and have intrinsic spatial dispersion and intrinsic temporal dispersion. A method to quantify the discretization error of an FDTD scheme is developed and is used to compare the errors of various schemes.

In lossy media, the relations between numerical phase and loss constants are derived for Yee's FDTD, ADI and the Crank-Nicolson-based methods, and verified with good agreement with numerical experiments. The numerical loss constant is always larger than its physical value, which implies that the electric field strengths computed by the FDTD methods in lossy media are smaller than the actual physical values. The numerical velocity in lossy media can be smaller or larger than its physical value.

The finite-difference operators and the efficient splitting scheme proposed in the thesis are powerful tools in developing new FDTD methods.

Acknowledgement

At this moment, I would like to acknowledge all the people who have helped me in the course of my Ph.D research. Great thanks of course go first to my supervisor, Dr. Trueman, who led me to the field of the finite-difference time-domain methods. It was with his full support, encouragement, critical review, and constructive comment and suggestion that I finally finished my research. Several times I thought it was too difficult to solve the problems. With his encouragement and trust, I finally found the solutions.

It is my pleasure to acknowledge Dr. Paknys, who taught my first course in Microwave Engineering at Concordia, and from him I learned how to find something behind the mathematics, though this thesis contains many mathematical operations. I thank Dr. Kubina for his encouragement and help. I appreciate Dr. McFee of McGill University, from whom I learned the finite element method. Special thanks to Mr. David Gaudine for his maintaining a ready-to-work computer facility and giving expert advices, Dr. Don Davis for his encouragement and Ms. Pam Fox for her great support.

I am grateful to my fellow students with whom I have spent good times: Dr. Hao Liu, MASc's Mr. Lei Wang, Mr. Chun Zhang, and Ms Weibo Sun; Ph.D candidates Mr. Ibrahem Abdulla, Mr. Guillaume Girard, Mr. Armin Parsa, and MASc candidates Mr. Monwar Hossain and Mr. Lixin Dou.

I appreciate the professors in my examining committee and the external examiner for their time and reviews.

Finally, I must express my gratitude to my wife and daughter for their understanding and support.

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

1D, 2D, 3D	one-, two- and three- dimensions
ABC	absorption or absorbing boundary condition
ADI	alternating (alternate) direction implicit
AOM	axes-optimized method
BC	boundary condition
CEM	computational electromagnetics
CFL	Courant-Friedrich-Levy
CM	coefficient modification technique
CN	Crank-Nicolson
CNAD	Crank-Nicolson approximate decoupling method
CNADDG	Crank-Nicolson approximate decoupling Douglas-Gunn method
CNCS	Crank-Nicolson cycle sweep method
CNCSU	Crank-Nicolson cycle-sweep-uniform method
CNDG	Crank-Nicolson-Douglas-Gunn method
CNDS	Crank-Nicolson direct splitting method
CPW	cells per wavelength
DOM	diagonally optimized method
EI	explicit/implicit
EM	electromagnetic
EMC/EMI	electromagnetic compatibility and electromagnetic interference
Eqn	equation
Eqs	equations
FD	finite difference
FDE	finite difference equation
FDE	finite-difference equation
FDTD	finite-difference time-domain
Fig	figure
HAM-FW	high-accuracy method with fixed weight
HAM-S	high-accuracy method by search
IE	implicit/explicit
IOM	isotropic optimization method
ITND	intrinsic temporal numerical dispersion
LHS	left-hand-side
MADM	minimum-average-dispersion method
ME	Maxwell's Equations
MEMS /NEMS	micro-electro-mechanical system /nano- electro-mechanical system
MKS	meter, Kelvin and second.
MPNE	maximum percent numerical error
MRTD	multisolution time-domain
NA	neighborhood averaging method
NA24	neighborhood-averaging (2,4)-stencil method
NAI	neighborhood-average isotropic method
NLPE	numerical loss percentage error
NVPE	numerical velocity percentage error

O24	optimized (2,4) stencil method
PDE	partial differential equation
PEC	perfect electric conductor
RHS	right-hand-side
RMS	root-mean-square
USM	unconditionally-stable method
w.r.t	with respect to

Symbols

L_o	linear differential operator for a partial differential equation (PDE)
L_s	linear spatial differential operator
L_b	linear operator for boundary
$U, U(x,y,z,t)$	function in a PDE to be determined
G	known excitation function
L_D	discretized linear differential operator
E_D	local discrete solution operator
λ	eigenvalue in Section 2.1, wavelength of a signal otherwise
M	linear operator for eigenvalue problem in Chapter 2
$u, u(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$	discretized function U to be solved
$u^n, u^n(i,j,k)$	simplified notation for u and $u(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$
g	discretized known excitation function
i, j, k	integer numbers for spatial indexing
k	the theoretical wavenumber in Chapter 9
n	time step index
Δx , Δy , Δz	spatial increments, cell sizes along x, y, and z coordinate axes
Δt	time step size
τ	local truncation error
$O(\Delta x^m)$	principal part of the truncation error
m	an integer in Chapter 2
λ	wavelength of a signal
f	time frequency of a signal
$\omega = 2\pi f$	angular frequency
$\vec{J} = \sigma \vec{E}$	conduction current
$J = \sqrt{-1}$	imaginary unit
ϕ	direction of wave travel with respect to x-axis (azimuth angle)
θ	direction of travel with respect to z-axis (elevation angle)
$\beta_x = \beta\cos(\phi)\sin(\theta)$	components of phase constant
$\beta_y = \beta\sin(\phi)\sin(\theta)$	
$\beta_z = \beta\cos(\theta)$	
$\beta = \sqrt{\beta_x^2 + \beta_y^2 + \beta_z^2}$	numerical wave phase constant
Q	amplification matrix

$\vec{E}, E_x, E_y, E_z$	electric field (volts/meter)
$\vec{H}, H_x, H_y, H_z$	magnetic field (amperes/meter^2)
$\vec{D}$	electric flux density (coulombs/meter^2)
$\vec{B}$	magnetic flux density (webers/meter^2)
$\vec{J} = \vec{J}_{source} + \sigma\vec{E}$	electric current density (amperes/meter^2)
$\vec{M} = \vec{M}_{source} + \sigma^*\vec{H}$	equivalent magnetic current density (volts/meter^2)
ε	Permittivity
μ	Permeability
σ	Conductivity
σ_m	equivalent magnetic loss (set to be zero in this thesis)
a_1, a_2	coefficients in FDTD update equations
$a_0 = (1-\sigma\Delta t/2\varepsilon)/(1+\sigma\Delta t/2\varepsilon)$	coefficients in lossy FDTD update equations
$a_{xe}, a_{ye}, a_{xh}, a_{yh}$	modified coefficients in lossless FDTD update equations
a_{cm}	modified coefficient for a square or cubic mesh
$P(\xi)$	amplification factor polynomial
$r_x = c\Delta t \sin(\beta_x \Delta x/2)/\Delta x$	plane wave coefficient of a Fourier mode
$r_y = c\Delta t \sin(\beta_y \Delta y/2)/\Delta y$	plane wave coefficient of a Fourier mode
$r_z = c\Delta t \sin(\beta_z \Delta z/2)/\Delta z$	plane wave coefficient of a Fourier mode
$p_{x,}, p_{y.}$	partial plane wave coefficients for the optimized method
(p,q)	pth order accurate in time and qth order accurate in space
p	desired numerical dispersion accuracy in Chapter 7
M_x, M_y, M_z	largest cell indices along x-, y- and z- coordinates
$\Delta_{\max} = \max\{\Delta x, \Delta y, \Delta z\}$	maximum cell size
$\Delta_{\min} = \max\{\Delta x, \Delta y, \Delta z\}$	minimum cell size
$s = c\Delta t/\Delta_{\max}$	Courant number
$s_c = 1/\sqrt{2}$	2D Courant limit
$R = \sqrt{(\Delta_{\max}/\Delta x)^2 + (\Delta_{\max}/\Delta y)^2 + (\Delta_{\max}/\Delta z)^2}$	mesh aspect ratio
$d = 1,2,3$	dumber of dimensions for 1D, 2D and 3D
W	vector containing discretized field components
w, w_{24}, w_{na}	weight parameters for the optimized methods
I	identity matrix, 3 by 3 in 2D,6 by 6 in 3D
D_1	difference matrix containing positive difference operators
D_2	difference matrix containing negative difference operators
c	Speed of light in free space
u	numerical phase velocity
u_p	light speed in lossy medium
u_m	numerical velocity at a desired accuracy p

u_r	relative numerical velocity, numerical dispersion
$\eta = \sqrt{\mu/\varepsilon}$	intrinsic impedance
$\tan\delta_c = \sigma/(\omega\varepsilon)$	physical loss tangent
$\tan\delta = ((\omega\Delta t/2)/\tan(\omega\Delta t/2))\tan\delta c$	numerical loss tangent
A_u	anisotropy of numerical dispersion
A_l	anisotropy of numerical loss
$\alpha_0 = \omega\sqrt{\mu\varepsilon\sec\delta}\,\sin(\delta/2)$	physical loss constant
$\beta_0 = \omega\sqrt{\mu\varepsilon\sec\delta}\,\cos(\delta/2)$	physical phase constant
$\gamma = \alpha + j\beta$	numerical propagation constant
ϕ	phase difference between electric and magnetic field in lossy case
N	mesh density, defined as $\lambda/\Delta x$
ψ	Fourier mode in complex form, continuous and discrete
$b = c\Delta t/2$	half the distance of light travel in Δt
D_x, D_y, D_z	central difference derivative approximation for 1st-degree derivative
D_{2x}, D_{2y}, D_{2z}	central difference derivative approximation for 2nd-degree derivative
D_{3x}, D_{3y}, D_{3z}	central difference derivative approximation for 3rd-degree derivative
$T(k), T(\omega)$	general transfer function of a Fourier mode
w_x, w_y	spatial transfer functions of a Fourier mode
w_t	temporal transfer functions of a Fourier mode
$D_{E,x}, D_{E,y}$	general difference operators for electric field component
$D_{H,x}, D_{H,y}$	general difference operators for magnetic field component
e	normalized discretization error vector
e_D	discretization error vector
e_1, e_2, e_3	normalized discretization error vector for individual update equations
e_{rms}	root-mean-square discretization error
A_0	theoretical coefficient matrix
T_p	plane-wave decomposition matrix
$A_{p,0}$	theoretical plane-wave matrix
$A_{\cdot}, A_{p,\cdot}$	discrete coefficient matrix and plane-wave matrix
ΔA	factorization error matrix in Chapter 6, error matrix in Chapter 8
$c_\phi = \cos\phi$, $s_\phi = \sin\phi$	cosine and sine of angle ϕ in Chapter 8
$c_t = \cos(\omega\Delta t/2)$	cosine of $\omega\Delta t/2$ in Chapter 8
$s_t = e^{-J\omega\Delta t/2}$	complex exponential function of $\omega\Delta t/2$ in Chapter 8
$r_t = \sin(\omega\Delta t/2)/\left(s\sqrt{d}\right)$	simplified expression of the LHS of Yee's dispersion relation
$r_t = \tan(\omega\Delta t/2)/\left(s\sqrt{d}\right)$	simplified expression of the LHS of USM dispersion relation

Chapter 1 Advances, Limitations and Challenges in Finite-Difference Time-Domain Methods

Since the propagation, diffraction, scattering, penetration and interaction phenomena of electromagnetic waves are governed by the well-known Maxwell's Equations, solving Maxwell's Equations is fundamental and crucial for the manipulation of electricity in the design of electric devices. But except for several special cases, such as plane waves, cylindrical waves, and spherical waves, or wave propagation in some waveguides, resonance in some cavities, or wave scattering by certain simple objects, it is often not possible to obtain an analytical or explicit solution for the electromagnetic fields [1]. Progress in digital computer hardware and software has now made numerical solutions popular. The numerical solution of Maxwell's Equations is the main focus of computational electromagnetics (CEM).

All fields of electrical engineering, encompassing power engineering, control engineering, electronics, and communications involve electromagnetics. Therefore, applications of electromagnetic simulation and numerical solutions can be found in many disciplines [2-6], such as antenna design and analysis, computer chips and circuits design, electromagnetic compatibility and electromagnetic interference (EMC/EMI), microwave engineering, radar cross-section analysis and design, optics and optoelectronics, lasers and photonics, microelectro-mechanical sensors, biomedical engineering and bio-technology, stealth technology and automatic target recognition, physics-based signal processing and imaging, remote sensing, and subsurface sensing.

There are many methods in CEM [1-12], based on the differential, integral, variational, hybrid approaches or other forms. These methods can be classified into two main categories: the time-domain methods and the frequency domain methods. In recent times, there is increased interest in direct time-domain methods. Time-domain methods work better for wideband signature studies, are better suited for parallel processing, and provide better visual representations for understanding field interaction phenomena [3]. In particular, time domain methods are the only means to examine the causal property of the recently emerging metamaterials with negative refraction index and positive/negative group velocities, for example, in [13].

Originally proposed by Yee [14] in 1966, and named later by Taflove [4] [10], the "finite-difference time-domain" (FDTD) method, a direct full-wave Maxwell's Equations solver, is the most popular method for the solutions of problems in electromagnetics [4-6]. After Mur's efficient absorbing boundary condition [15] was introduced in 1981, the FDTD method gained wide acceptance, particularly in 1990s. This is because FDTD is quite versatile for solving Maxwell's Equations from extreme low frequencies to microwave and optical frequencies [4], and even for problems in particle physics [16]. FDTD is simple to program using elementary data structures, and has a relatively high computational speed. It can make broad-band prediction, deal with arbitrary 2D and 3D geometries with materials of any conductivity, permittivity and permeability, with nonlinear materials, with frequency- and time-dependent materials, as well as with unconventional materials such as anisotropic plasmas, magnetized ferrites and metamaterials [2-6] [13]. In addition, it is robust, flexible, and non-dissipative.

Many enhancements have been introduced to extend FDTD to new applications and improve the performance of FDTD, such as accuracy, computational efficiency, boundary treatment and material interfaces. For brevity, this Chapter will be restricted to advances related to the main topics of this thesis.

1.1 Applications of the Finite-Difference Time-Domain Methods

Initially, FDTD was invented for solving Maxwell's Equations in electrical engineering. Due to its superiority over other methods, the ingenious FDTD method has been adapted to other areas, such as acoustics [6], quantum mechanics[17], elastodynamics [18], as well as particle physics [16], to name a few.

The applications of FDTD methods have swept almost the whole spectrum of electromagnetics: static [19], quasi-static [20], lightning-interaction with aircraft [21], power transmission, antenna, microwave, wireless communications, and photonics [4].

FDTD can accurately predict transient behavior and steady-state behavior over a wide bandwidth of devices and components, such as high speed digital or mixed signal circuits, printed circuit boards, integrated circuits, multichip modules, MEMS and nano-components. In recent years, FDTD has been extended to system level. The progress of FDTD methods has made simulations possible for devices, components and systems, including passive and active components.

1.2 Limitations in FDTD Methods

In computational electromagnetics, the size of geometrical features and the wavelength inside the materials are very important parameters. The "electrical size" of an object [22], that is, the geometrical size scaled by the wavelength, is used as a guide. If the electrical size is much larger than unity, such as the analysis of an optical waveguide,

huge computer storage is required to solve the problem with FDTD. If the electrical size is very small, it is often over sampling spatially and thus spatially for explicit FDTD, because the Courant-Friedrich-Levy (CFL) constraint [4] (CFL limit, CFL condition or stability condition) specifies the maximum time step size for stability in terms of spatial sampling size or the "cell size". For example, the electrical size for integrated circuits is of order 10^{-5}. To simulate such objects, the CFL dictates a very small time step size, leading to a long CPU time. To understand how the CFL condition limits the FDTD performance, consider the interaction of a cell phone with human head as an example. If homogeneous (uniform) meshing is used, as many as 8.5×10^{9} cells with 1.5 TB memory may be needed, and may take spend 2.5 years to execute [23]!

Like other discrete methods, FDTD has numerical errors. The most important is the numerical dispersion, a phenomenon caused by the fact that the velocity of the numerical wave is not the same as the physical speed, causing numerical refraction and reflection [4]. Numerical dispersion causes phase error as the wave propagates through the FDTD grid, and depends on direction of wave travel, which is a limiting factor for electrically-large structures. In addition, FDTD has magnitude error [161*].

1.3 Methods for Improving Numerical Accuracy

To improve FDTD's numerical accuracy, many methods have been proposed. These methods can be classified into two main categories: methods using totally new formulations, such as the multi-resolution time-domain (MRTD) method [24] and the pseudospectral time-domain method [25]; and methods modifying the conventional Yee's method. Chapter 3 of this thesis focuses on the latter, which can be further classified into

* References from 151 to 167 are the published papers and internal reports by the author, listed on page 225.

three methodologies: larger computational stencil or "higher-order" schemes; coefficient modification schemes; and hybrid methods using a large computational stencil and coefficient modification.

Conventional Yee's FDTD is 2^{nd}-order accurate in both time and space, and is referred as a (2, 2) method. The (2, 4) scheme is 2^{nd}-order accurate in time and 4^{th} -order in space and can model structures that are thousands of wavelengths in size [26-30]. General (2, 2M) methods where $M \geq 2$ can be found in [18]. The (4, 4) schemes such as Rounge-Kutta [10] and dissipative scheme [31] have been proposed. A generalized higher-order (4,2M) scheme using the discrete singular convolution for spatial differentiation is reported [32]. Comparison of some higher-order methods can be found in [10] [33-35]. However, higher-order algorithms have degraded numerical stability properties relative to the second-order algorithm, which means that the time-step size must be smaller than that for second-order methods. Additional challenge for higher-order schemes is the proper treatment of boundary conditions and material discontinuities to eliminate artificial reflection and refraction, and avoid later time stability [10]. Most importantly, analysis shows that higher-order schemes do not always give better numerical accuracy [159].

Simple methods to improve numerical accuracy at one frequency either use artificial anisotropy [36-38] or modify the speed through the difference operators [39]. Hybrid methods include those using non-standard finite-difference method and larger stencils [40-43], overlapped lattice method [44], and filtering methods [45-48].

In addition, the Richard's extrapolation can be effectively used after running an FDTD code by halving the spatial cell size [49]. However, it is only effective for uniform discretization.

1.4 Methods Using Larger Time Step Size Than CFL Limit

The CFL time step size limit arises from stability requirements. Several methods can be used efficiently in analyzing the stability of FDTD methods [50-55]. Because the time step size in explicit FDTD imposed by the CFL limit is too small for practical computation for some problems, many methods for relaxing this time constraint have been proposed.

Some problems involve small parts such as a thin wire. If the problem uses uniform discretization everywhere, it is the smallest part that determines the finest cell size, which is only a small part of the whole volume but results in larger computation time and memory requirements. Sub-griding schemes are efficient in improving the CPU time consumption, and can be divided to three main categories: sequential computations or time-domain segmentation techniques [56]; sub-gridding in space [57]; and sub-gridding in both space and time [58-59]. The transition between the coarse and fine regions must be treated with care in order to eliminate artificial reflection at the interfaces [58].

By combining FDTD and reduced-order modeling, a sub-domain model in 2D is achieved [60] with time-step size larger than Courant limit, and more importantly, it reduces memory by a factor of 100, but at the cost of expensive computation.

For some narrowband application, the envelope method can be used [61-62], which in some cases can have no time step size limitation [63]. Extrapolation is also efficient in shortening the CPU time, such as finite-impulse-response neural network model which

uses a short segment of FDTD to extrapolate later information [64], Prony's method other signal processing methods [65] for certain problems.

Finally, to reduce computer run-time consumption, special hardware [66-67] or a parallel computer [68] can be used.

1.5 Methods Without the Time Step Size Limitations

It has been shown [69] that for hyperbolic systems of partial differential equations, there are no explicit, unconditionally stable, and consistent finite-difference schemes. Therefore the only way to eliminate the Courant condition is to seek implicit schemes, or unconditionally-stable methods (USMs).

1.5.1 No time stepping schemes

Most current FDTD methods use time-marching schemes to advance the time step by step. If the step-by-step time marching can be eliminated, there will be no time step size, hence no CFL constraint. This thesis refers such a scheme as a "no-time-stepping method". By transforming Maxwell's Curl Equations into the Laplace domain and combining with the standard FDTD method, a simple no-time-stepping method is realized for some waveguide problems [70]. By expressing the transient behaviors in terms of weighted Lagurre polynomials [71], a new unconditionally-stable method solves 2D problem in space only. Based on the Suzuki product-formula, Chebyshev polynomials can be used to describe the time evolution [72]. It is said these algorithms are unconditionally stable by construction [73]. Higher-order algorithms have also reported by the same group [74].

1.5.2. Alternating-direction-implicit methods

The Alternating-Direction-Implicit (ADI) method originally proposed by Peaceman and Rachford is a popular method in solving parabolic and hyperbolic differential equations [49][75-78]. Extensive literature survey on ADI by the present author can be found in [165]. Because it can use a large time step size, it can also be used in solving linear and nonlinear elliptic equations [79].

In solving Maxwell's Equations, the first ADI method with three alternations was introduced in 1984 [80]. In 1999 and 2000, this method was re-invented with two-alternations [81-83]. Since then the ADI-FDTD has been studied extensively, extended to frequency-dependent materials [84], lossy media [85], highly conductive materials [86], Debye material [87], incorporated into a sub-griding scheme [88-89], developed with compact schemes [90-91], and extended to cylindrical coordinate [92]. Reference [93] proposes the alternating-implicit block-overlapped (AIBO) FDTD method.

Various methods have been proposed to improve the numerical dispersion of ADI-FDTD, such as the higher-order scheme [94-97], the coefficient- modification methods [38] [98], and the envelope ADI-FDTD [61] [62].

However, ADI suffers from large anisotropy in addition to large numerical dispersion [152-154], and the numerical dispersion relations in the literature are not consistent with each other [81] [99-101]. The author first found this problem [166] [160], and gave a numerically-verified relation in 1D [160] [154] and 2D [152]. Later other researchers published similar results [102-103].

1.5.3. Crank-Nicolson scheme based methods

The Crank-Nicolson scheme is well known in computational science and is a high-accuracy method for solving differential equations [49] [75-78]. But the resulting matrix is block-tridiagonal or tridiagonal-with-fringes in 2D and 3D, which is very expensive to solve by usual methods, such as Gaussian elimination or iterative methods. However, the CN scheme can be used to compare the accuracy with other USMs, for example the ADI-FDTD [104]. The first practical use of the CN scheme in solving Maxwell's Equations was used in a parallel computation by decomposition of the eigenvalue/eigenvector for the wave equations of the second degree [105]. A domain-decomposition method [68] first solves a Schur complement system, then solves a block tridiagonal matrix in 2D. Reference [16] uses CN scheme for solving particle problems using ADI algorithm. Reference [106-107] use the split-step method that decouples the 3D Maxwell's Equations into locally 1D equations, which is similar to the characterization method [108-110].

1.5.4. Other unconditionally-stable methods

Characteristic-based algorithms [108-109] first decouple Maxwell's Curl Equations into 1D equations [77], then solve them implicitly. Recently a two-factor scheme by combining a characteristic-based approach to spatial differencing with an implicit lower-upper approximate factorization is proposed in 1D, which avoids the solution of a tridiagonal system [110].

1.6 Other Advances

The regular Yee mesh is an orthogonal, rectangular, structured grid, and uses the same cell sizes over the whole computational domain. Such a meshing method causes

"staircase error" in the approximation of curved boundaries. In addition, traditional uniform FDTD has difficulty dealing with discontinuities such as edges and corners involving dissimilar media [111]. To alleviate the error due to boundary discretization, locally conformal, irregular-mesh (including non-orthogonal grid, curvilinear) methods and many hybrid schemes such as hybridizing with non-structured finite element method in the time domain have been proposed [3-4] [10] [111]. Recently a staggered upwind embedded boundary method has been proposed to eliminate the staircase error [112].

An early attempt to reduce the memory requirement is to use the divergence-free electric-field regions, and combine the scalar wave equation [113]. By using the vector potential formulation, a 33% memory reduction is achieved for 2-D problems [114]. Very recently, the reduced FDTD [115] method has been proposed to eliminate the necessity of subdividing regions, maintaining the advantage of four required field components, while also being able to easily treat conductors and source regions by using the divergence-free nature of the electric displacement instead of the electric field, and following a specific sequence for the spatial update of the remaining field components. This method achieves a 33% memory reduction in 3D.

Using a discrete Green's function formulation of FDTD method [116-117] is also an efficient way to save memory. For antenna radiation problems, it requires neither absorbing boundary conditions nor the modeling of free space nodes. For some special cases, three-dimensional problems can be simplified to two dimensions (compact scheme), which can reduce greatly the required memory space and CPU time [3].

When applying the FDTD technique to open radiation problems, an accurate and computationally-efficient absorbing boundary condition (ABC) must be developed in

order to truncate the mesh lattices. The ideal ABC is reflectionless over a broad bandwidth. The most commonly-used methods are Mur's [15], Liao's [118], and particularly the Berenger's PML [4] [21] [119]. The PML is composed of a non-physical absorbing material and is placed adjacent to the boundary cells of the FDTD grid. Its wave impedance is independent of the angle at which the scattered waves impinge on it. The main advantage of the PML is its ability to maintain a high level of performance over a wide range of group velocities, and its performance is independent of the dispersive nature of the propagating medium. However, each new FDTD method requires its own ABC-PML to be compatible to the FDTD scheme in accuracy or other aspects.

1.7 Motivations and Challenges

From above literature survey it can be seen that, although FDTD methods have matured in many applications, and have been extended to increasingly-complicated problems, there are some limitations. Sometimes the limitations are so severe that it is very difficult, or very inefficient to apply FDTD methods to certain problems. For example, electrically very-large structures are difficult to model with FDTD because of phase error accumulation; for objects with very fine geometrical features compared to the wavelength, the CFL constraint limits the largest time step size to be so small that the conventional explicit FDTD methods require prohibitively long CPU time to simulate.

Since explicit FDTD methods are more efficient and easy to implement for many problems, one goal of this thesis is to develop new explicit methods that have higher numerical accuracy. The challenge is that, with a history of about 40 years, explicit FDTD methods have been investigated thoroughly. To devise a new algorithm, new

methodology must be sought, and inspiration from other computational methods is necessary.

Another main purpose of this thesis is to develop some unconditionally-stable methods to overcome the CFL constraint and ADI-FDTD's large anisotropy error. Though some CN-based methods have been proposed [104-106], they either require much more computer resources than explicit FDTD methods, or the numerical dispersion relation of the method is not given, therefore it is difficult to design a mesh model to achieve a pre-set numerical accuracy. The big challenge is to avoid an expensive solution of a block-tridiagonal matrix, in particular for the first-degree Maxwell's Equations which have three coupled field components in 2D and six in 3D.

An unconditionally stable scheme is one in which the time step size is not bounded by any stability requirement. But the time step size may be governed by other factors. One obvious bound is the Nyquist Criterion, which governs all discrete systems. However, the Nyquist Criterion has not attracted the attention of the FDTD community, possibly because previous explicit FDTD methods discretize much more densely than the criterion.

Note that most memory-saving methods and staircase-error-free methods modify the update equations of FDTD methods. Therefore, this thesis will focus on developing new FDTD methods with Yee's mesh. The enhancement of the methods in this thesis for memory saving and the elimination of staircase errors will be left for future investigation.

FDTD methods can be applied to lossy materials, and several papers investigated the numerical dispersion relations [120-124]. However, most authors omit some terms in their analysis, which are crucial to obtain the correct numerical accuracy information in

lossy media. Therefore the numerical dispersion relation in lossy material is re-investigated in this thesis.

At present, only numerical dispersion is evaluated to assess a FDTD method in literature, which accounts only the phase error. However, the magnitude error must also be quantified, particularly for problems in which the magnitude is an important parameter, such as in EMC and bio-electromagnetics. This thesis will develop such a quantification method [161].

1.8 Contributions and Problems to Be Solved

This thesis focuses mainly on developing new methods that either improve numerical accuracy or remove the CFL limit. The contributions fall into four areas. First, some new methods are proposed, including optimized explicit methods with high accuracy and unconditionally-stable methods with better performance. Second, numerical loss is found in lossy media associated with discretization but generally neglected by FDTD community. Third, a new discretization-error quantification method is presented that reveals more information than the usual numerical dispersion relation.

A deep understanding of the fundamental characteristics of the FDTD methods allows better methods to be invented, and a good choice of an FDTD method to be made for a specific problem. This thesis anticipates providing some such understanding. For example, it has been observed that, there are only two propagating modes and no stationary mode in 1D, two propagating modes and one stationary mode in 2D, and four propagating modes and two stationary modes in 3D for a stable FDTD scheme. The reason for this behavior has not been reported. This thesis will explore some basic questions like this, such as the mechanism of unconditional stability for ADI-FDTD,

intrinsic numerical dispersion, and various time step size limits applied to the unconditionally-stable methods.

Another goal of this thesis is to ensure that the theoretical assumptions that have been made in analyzing FDTD methods are not deficient, and that no mechanism has been neglected. This is achieved by numerical validation of the numerical dispersion relation of the proposed methods.

In developing new schemes, the concept of spatial difference operators gradually comes into mind. With some trial and error, as well as comparison with the traditional node-based formulation, it is realized that, like continuous differential operators, the difference operators can be treated as a mathematical symbol. The mathematical operations for such operators are the same as algebraic variables. Thus the derivation of many formulas becomes much easier, and the update equations are more compact and concise. The use of difference operators also shortens the length of this thesis which allows writing concisely the formulas. In addition, the results derived using the difference operators can be directly extended to higher-order methods by simply using higher-order difference formulas.

1.9 Outline of the Work

This thesis is outlined as follows. Chapter 2 introduces some terminology and fundamentals of the FDTD methods. In addition, the method to extract numerical velocity from numerical experiments is described.

Chapter 3 proposes some optimized explicit methods with high accuracy. Instead of pursuing higher-order schemes, this thesis seeks methods with less numerical dispersion by minimizing dispersion error in various senses. By the use of the standard (2, 4) stencil,

the optimized O24 method [159] can have much less numerical dispersion in a pre-designed sector of wave travel, or can eliminate anisotropy completely at one designated frequency and reduce anisotropy at other frequencies. By employing an average around the neighborhood of a field component, the neighborhood-averaging (NA) method can achieve high numerical accuracy with time step sizes larger than the Courant limit. A more promising method is the neighborhood-averaging 24-stencil (NA24) method [164], which has one order-of-magnitude higher accuracy than the methods reported in [41] [44] with the Courant limit time step size in a wide bandwidth. In addition, it reveals that the standard (2,4) method is only 2^{nd}-order accurate in terms of numerical dispersion.

Chapter 4 first addresses the Crank-Nicolson scheme for discretizing Maxwell's Equations, then formulates the ADI method in a matrix form, and discusses the numerical dissipation and growth in each individual sub-procedure of the ADI-FDTD method [152] [154] [160]. The numerical dispersion is re-derived and validated from numerical experiments, and the inconsistency among the published papers is removed. In addition, an isotropic ADI method with the (2, 4) stencil is proposed, which has no anisotropy at one frequency, and greatly reduces the numerical dispersion at the same time. This is a surprising result, and is different from its explicit counterpart. The significance of this chapter is to reveal that is the exact cancellation of the growth and dissipation in the two sub-marching procedures that makes the overall ADI-FDTD method stable. In addition, the perturbation errors of ADI-FDTD relative to the CN scheme are obtained, which are of the 4^{th}, 3^{rd} and 2^{nd} order.

Chapter 5 presents several new 2D unconditionally-stable methods based on the CN scheme. Different from the ADI method, the intermediate time step in these methods is

not associated with any specific time, making the enforcement of the current- or voltage-source values more effective. The stability, numerical dispersion and perturbation errors are analyzed, and numerical dispersion behaviors are discussed and compared to ADI method [151] [153] [157] [167]. Most CN-based methods have much smaller anisotropy than ADI, which is inherited from the original CN formulation [153]. In addition, a non-dissipative, unconditionally-stable method for solving the wave equation of the second-degree is proposed.

Chapter 6 focuses on the unconditionally-stable methods for solving 3D Maxwell's Equations, which is more challenging than 2D because there are six coupled field components. With the use of difference operators and decomposition of the related matrices, several different methods for efficiently implementing the CN scheme have been proposed, such as the approximate-factorization-splitting method [156] and cycle-sweep method [163]. The stability is analyzed for all the schemes, the numerical dispersion relations are given, and the perturbation errors are derived.

Chapter 7 discusses some fundamental characteristics of the unconditionally-stable methods, such as intrinsic spatial-derivative-related dispersion, intrinsic temporal numerical dispersion, time-step-size limit due to numerical attenuation, and numerical accuracy [153-154]. In particular the Nyquist Criterion is emphasized in analyzing some properties of FDTD methods. It is shown that to avoid numerical attenuation, these unconditional-stable methods do have an upper-bound for the time step size, which are not originated from the stability requirement [153-154]. Classification of the problems suitable for unconditionally-stable methods allows the choice of proper method for a

specific problem, and a guide to choose the time step size is given to achieve a desired accuracy [155].

Chapter 8 introduces a new concept of numerical loss in FDTD to reveal that in lossy media, the numerical loss constant is not the same as the physical loss constant, due to discretization. This is logical and mathematically correct, because the loss constant and the phase constant together make up the propagation constant. If one has numerical error, the other can also suffer from error. The general dispersion and loss relations for Yee's FDTD, ADI-FDTD and CN-FDTD in 1D are discussed, and numerically verified. It is shown that CN-FDTD has higher numerical accuracy than ADI. In particular the error of the numerical loss constant is much smaller than that of ADI-FDTD. The perturbation error of ADI-FDTD is 1^{st}-order accurate in time, leading to larger numerical loss error.

Chapter 9 provides a quantitative method to evaluate the magnitude error or discretization error, which provides more information of a scheme, and allows a deep insight into the numerical errors. After normalizing and decomposing the plane wave, Maxwell's Equations in the continuous domain and in the computational domain can be written in matrix forms. The difference between the matrices in the discrete domain and the continuous domain reveals the discretization error. Each update equation has its own discretization error and is different from the error of other update equations. Since the information is overloaded, the root-mean-square (RMS) error is used to measure the discretization error. The discretization errors for Yee's FDTD, for the optimized explicit methods, for ADI-FDTD and for CN-based FDTD methods are given and discussed.

Chapter 10 summarizes the progress achieved in this thesis, gives some observations for insight into some issues of the FDTD methods, and indicates future work.

Chapter 2 Fundamentals of Finite-Difference Time-Domain Methods

The study of FDTD relies to a great extent on applied mathematics, electromagnetics and electrical engineering. This chapter describes some useful terminology and fundamentals about numerically solving partial differential equations (PDEs) and Yee's FDTD. The description is brief and not intended for a strict and complete introduction in terms of mathematics.

2.1 Partial Differential Equations

Many field problems, physical and natural phenomena, such as electromagnetic, acoustic, heat, ocean, seismic, meteorological, solid geophysical, and fluid mechanic waves, can be described with PDEs. PDEs form the basis of very many mathematical models of physical and biological phenomena, and their use has been spread into economics, financial forecasting and other fields. If a PDE contains only first-degree algebraic terms in the relevant variables, it is linear and obeys the principle of superposition.

The elliptic, parabolic, and hyperbolic PDEs are deterministic as opposed to stochastic, and can be described with $L_o U = G$, where L_o is a linear operator, G is a known excitation source, and U is the function to be determined. An eigenvalue problem is non-deterministic in which G takes the form of λMU where λ is the eigenvalue, and M is a linear operator. A unique solution of a PDE can be obtained only when boundary conditions are specified for the problem. The boundary conditions (BCs) are described with $L_b U = B$ [77], where L_b is a linear operator. When the BCs are given at time $t = 0$,

the problem is called initial-value (Cauchy) problem. If only the values at geometrical boundaries are specified, the problem is a boundary-value problem [77]. The periodic boundary condition is another important one that extends the associated initial value function B periodically to the whole real space.

The solution region can be either closed (bounded, interior) with given boundary conditions such as a resonator, open (unbounded) which extends to infinity such as antenna radiation problems, or mixed (hybrid) such as an open-end waveguide.

2.2 Discretization, Mesh and Update Equations

In order to be suitable for numerical computation, it is necessary to approximate the solution of the PDEs by discretizing the space and time. Such discretization transforms PDEs from the continuous domain into the discrete, "computational domain" [126]. After such discretization, the differential equation becomes a difference equation $L_D u = g$, where L_D is discretization of the linear operator [126], u is the discretized, approximate solution of U, and g is the discretized G. The discretization takes different forms for various numerical methods. In finite difference method, the unknown function $U(x,y,z,t)$ is approximated as $u(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$ where i, j, k, are the spatial indices, n is the time step index, and Δx, Δy, Δz and Δt are spatial and temporal increments. In this thesis, the temporal increment is termed the time step size. $u(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$ can be written more simply as $u^n(i,j,k)$ [77], and sometimes just denoted as u^n.

In the continuous domain, the unknown quantity can be computed at any time and any location. However, in the computational domain, numerical methods can only compute such unknown quantities at the specified, isolated "points" or nodes. Usually, the region to be examined is divided into a rectilinear grid with sides parallel to the

spatial axes. Thus the computational domain is a regular mesh, or grid (or net, network, pattern, lattice, tessellation) [126-127] which is composed of small bricks of size Δx by Δy by Δz, which are also called cuboids (3D), rectangles or squares (2D), and segments (1D). Each of these small elements is called a cell. There are irregular meshes, such as triangles, hexagons [127-128], curvilinear net [4] [127]; they can be in rectangular, cylindrical and spherical coordinates [10]. There are also irregular nets called curvilinear meshes. The meshes can also be classified as orthogonal and non-orthogonal [4].

In actual computation, the difference equation $L_D u = g$ is expressed with local discrete solution operator E_D that $u^{n+1}(i,j,k) = E_D u^n(i,j,k)$ [126]. This form of equation shows the evolution of the unknown quantity at time step n+1 from known quantities at previous time steps (n, n-1 etc.), and usually called an "update equation". A specific discrete solution operator is referred as a scheme, or a method, and sometimes an algorithm. The grid nodes involved in the computation of E_D are called the computational stencil, molecule or foot-print [127] [129].

2.3 Finite Difference Formulas and Local Truncation Error

In finite difference methods, the Taylor series expansion is used to analyze the approximation. For example,

$$U(x\pm\frac{\Delta x}{2}) = U(x) \pm U_x \frac{\Delta x}{2} + \frac{1}{2}U_{2x}(\frac{\Delta x}{2})^2 \pm \frac{1}{6}U_{3x}(\frac{\Delta x}{2})^3 + \frac{1}{24}U_{2x}(\frac{\Delta x}{2})^4 \pm \frac{1}{120}U_{3x}(\frac{\Delta x}{2})^5 + \ldots \qquad (2\text{-}1)$$

where U_{ix} (i=1,2,...) are the first, second, and higher-degree derivatives. Symbolically a difference equation can be formulated in exactly the same way as the differential equation using an exponential function [127] [130], in which U^{n+1} can be expressed as $\exp(\Delta t L_s)U^n$ for the PDE $\partial U / \partial t = L_s U$ where L_s is a linear spatial differential operator.

In practice, the exponential function must be simplified by using the Taylor expansion, which is truncated to include only some lower-order terms. The numerical computation can use only a *finite* number of terms from the Taylor expansion, thus it is called the finite difference method. The resulting discrete equation is the finite difference equation (FDE) or simply the difference equation. The finite difference method introduces error due to approximation, usually called truncation error, or discretization error [49] [127] [130].

The first derivative can be written from Eqn. (2-1) as

$$U_x = \frac{U(x+\Delta x/2) - U(x-\Delta x/2)}{\Delta x} + \tau \tag{2-2}$$

$$\tau = -\frac{1}{24} U_{3x} (\Delta x)^2 - \frac{1}{19200} U_{5x} (\Delta x)^4 - \ldots \tag{2-3}$$

where U_{3x}, U_{5x} are 3rd - and 5th -degree derivatives, respectively. Usually τ is called the local truncation error [130], or discretization error [49] to account for the case that some functions may have no higher-degree derivatives. The first term in τ is referred as the principal part of the truncation error. For brevity, the big O notation is commonly used to denote the principal part as $O(\Delta x^m)$, where m is an integer[76]. The error in Eqn. (2-3) can be written as $O(\Delta x^2)$, since the lowest order of Δx is two. The quantity before $(\Delta x)^m$ is called transformation coefficient.

Since the finite difference method can only use finite number of terms, Eqn. (2-2) may be approximated to

$$U_x(x) \approx u_x(x) = \frac{u(x+\Delta x/2) - u(x-\Delta x/2)}{\Delta x} \tag{2-4a}$$

provide that τ is bounded. Similarly, the second derivative can be approximated to

$$U_{2x}(x) \approx u_{2x}(x) = \frac{u(x+\Delta x/2)+u(x-\Delta x/2)-2u(x)}{\Delta x} \tag{2-4b}$$

The computational stencils for some finite difference formulas can be found in [11] [18] [33] [127]. Eqs. (2-4) and (2-5) are referred central-differencing formulas [11] [127]. Most FDTD methods use central differences.

2.4 Convergence, Accuracy, Stability, Nyquist Criterion and Consistency

To evaluate a numerical scheme, the convergence, accuracy, stability and consistency must be analyzed [49] [69] [129-130]. "Convergence" means that as the discrete spatial increments and time step size approach zero, the solution of the difference equation approaches the solution of the original differential equation, that is $u|_{\Delta x,\Delta y,\Delta z,\Delta t\to 0} \to U$ at any point inside the computational domain. "Global accuracy" (the set of values $\{u^n(i,j,k)-U^n(i,j,k)\}$ [130]) is a measure of the difference between the numerical solution and the exact solution all over the region under consideration, and is generally a very difficult quantity to estimate [130]. To compare different schemes, the order of accuracy is usually used. A difference scheme is said to be accurate of order (p,q) to the given PDE if the local discretization error $L_D U^n(i,j,k) - G^n(i,j,k)$ or $\tau = (u^{n+1} - E_D u^n)/\Delta t$ of the difference scheme can be expressed as $\|\tau\| = O(\Delta t^p) + O(\Delta x^q)$ [49] [69] [77] [126]. Strictly it is local order of accuracy since it does not concern the boundary conditions. Note that τ is not the error of the solution. For smooth initial data, the order of the solution is equal to the order of accuracy of the scheme [69]. Several methods can be used to improve the accuracy of a solution [49].

In addition, the difference equations may not be solved exactly because of finite precision of machine arithmetic, and the introduced error is called round-off error [127].

In linear PDEs, the round-off errors are cumulative [130]. They cannot generally be reduced by decreasing the cell sizes [127] [130]. Round-off error reflects the fact that computations must be done only with finite precision arithmetic on a computer [11]. Decreasing truncation errors by using a finer mesh may result in increasing the round-off error due to the increased number of arithmetic operations [11].

The stability of a finite difference scheme requires that the difference $u-\tilde{u}$ between the theoretical solution u with infinite-precision arithmetic and the actual solution $\tilde{u}$ with finite-precision arithmetic remain bounded as n increases with the time step size fixed [130]. A numerical algorithm is said to be stable if a small error at any stage produces a smaller cumulative error [11]. An algorithm is stable if the local discrete solution operator is stable at any time and at any point [126]. A more practical requirement for stability is $\|u^{n+1}\| \le \|u^n\|$, or $\|E_D\| \le 1$. Note that the stability described above usually concerns a time slot from 0 to $t_{\max}$. Such stability is refereed to numerical stability [69], and will be examined in this thesis. In addition, this thesis does not distinguish the solutions u and $\tilde{u}$ for convenience since all the schemes presented in this thesis are stable.

Modern research shows that the above conventional description of stability is weak, and not sufficient [51]. Improperly posed material interfaces have a more restrictive stability requirement [55], and may cause instability, particularly for higher-order schemes [54]. Reference [55] shows that if the material properties are averaged on the boundary, the more restrictive condition can be avoided, and thus the stability of the original scheme is not affected. Since this thesis does not involve the treatment of boundary conditions, the conventional stability analysis is used.

While accuracy analysis gives the order of accuracy, and the stability analysis gives how big the temporal increment can be for a given mesh density, the Nyquist criterion guides the biggest temporal and spatial increments such that the signal can still be recovered correctly [131]. The Nyquist criterion is $\Delta t \leq 1/2f$ for the time step size where f is the highest frequency of the signal, and $\Delta x \leq \lambda/2$ for 1D spatial cell size where λ is the shortest wavelength of the signal. References.[153-154] pointed out that ADI-FDTD and Crank-Nicolson (CN)-based schemes cannot reach the Nyquist time sampling limit; Reference [132] concludes that Yee's FDTD cannot use the coarsest spatial sampling $\beta\Delta x = \pi$ at the "magic time step" [4]. Hence in determining the stability, only $0 \leq \beta\Delta x < \pi$ and $0 \leq \omega\Delta t < \pi$ are considered. Use of larger increments than the above range may lead to a "fake" or misleading conclusion concerning stability.

Consistency (or compatibility, [49]) requires that $\tau/\Delta t \to 0$ as spatial increments go to zero [130]. If a finite difference scheme is consistent, the difference equations are good approximations to the PDEs [77], and the numerical solution is an approximate solution of the original differential equation [130]. For a consistent scheme, stability is the necessary and sufficient condition for convergence [127]. In addition, the Lax Equivalence Theorem [77] states that a consistent, two-level (see section 2.7) difference scheme of a well-posed linear initial-value problem is convergent if and only if it is stable. All the schemes presented in this thesis have the order of accuracy (2, 2) that ensures the consistency. Thus we will only analyze the stability issue for such schemes.

If a scheme is stable for a lossless material, it is stable for a lossy material as well [69]. Thus, all the schemes (which are proved being stable) in this thesis can be extended to lossy cases without having to prove their stability. In the lossy case, the time step size

should be small enough to resolve the medium relaxation time $\tau_r = \varepsilon/\sigma$ [120]. In addition, in lossy materials, Maxwell's Equations are dynamically stable as time extends infinitely [69].

2.5 Fourier Analysis Method and Matrix Method

There are many methods for the analysis of stability [129], but a convenient and popular approach is the Fourier analysis method (or von Neumann method) [4] [52][128]. The numerical solution can be harmonically decomposed into the Fourier modes, and a Fourier mode (sometimes called grid function [69]) is an exact solution of the difference equations [76]. Instead of finding the difference $u - \tilde{u}$, the Fourier method introduces the amplification factor (or symbol, [75]) ξ together with one Fourier mode, because of linearity. A Fourier mode for the lossless case can be written as

$$u^n = \xi^n \exp\left(-J(\beta_x i\Delta x + \beta_y j\Delta y + \beta_z k\Delta z)\right) \tag{2-5}$$

where $\beta_x = \beta\cos(\phi)\sin(\theta)$, $\beta_y = \beta\sin(\phi)\sin(\theta)$, $\beta_z = \beta\cos(\theta)$, and $\beta = \sqrt{\beta_x^2 + \beta_y^2 + \beta_z^2}$ is the numerical wave phase constant; ϕ is the direction of wave travel with respect to x-axis (azimuth angle), and θ is the direction of travel with respect to z-axis (complimentary to elevation angle). Then $u^{n+1} = \xi u^n$. The amplification factor $\xi = u^{n+1}/u^n = u^{n+1/2}/u^{n-1/2}$ can be expressed as $\xi = e^{-J(\omega_r + J\omega_i)n\Delta t}$ [49], which describes the time evolution of the function in the computational domain. For the subsequent time steps, when the magnitude of the amplification factor is larger than unity, the scheme exhibits "growth" with time. If the magnitude is smaller than unity, then there is "dissipation". When the magnitude is exactly unity, the method is "strictly non-dissipative" [69], "neutral" or "marginally stable". A scheme is stable only when $|\xi| \le 1$. If

neither growth nor dissipation represents physical behavior, a stable FDTD scheme requires $|\xi| = 1$. Note that generally the phase constant in a scheme is different from its theoretical or physical value. This is called discrete dispersion [77] or "numerical dispersion" [4]. Using Fourier analysis can also show the order of accuracy [69] [160].

Note that the Fourier method does not consider BCs since it is effective in an unbounded uniform Cartesian grid. The matrix method is well suited to include BCs. It finds the eigenvalues of the amplification (or polynomial coefficient [129]) matrix Q ($Qu^n = 0$) from the update equation including the BCs. It can also be simplified by using a Fourier mode without concerning the BCs. In this case, it leads to the same result as the Fourier method. If all the eigenvalues, in absolute value, are less than or equal to 1, the scheme is stable [127]. This method is applicable to linear PDEs with non-constant coefficients [130]. If the amplification matrix is symmetric, the Fourier analysis is necessary and sufficient to ensure stability [77]. The necessary and sufficient condition for the difference equations to be numerically stable when the solution of the partial differential equation does not increase as t increases is that the Lax-Richtmyer definition of stability be satisfied $\|Q\| \le 1$.

The determinant of Q (whether containing the boundary conditions or not) is the amplification (characteristic, or associated stability) polynomial of order m (1D, m=2,2D, m=3, 3D, m=6 generally) [69]. By finding the roots of the polynomial, the amplification factor ξ can be obtained. A stable scheme is said to have a "conservative" amplification polynomial if $|\xi| = 1$ [69] [129], where the locus of the amplification factor lies on a unit circle. The requirement $|\xi| = 1$ assures that the energy in different modes will not be incorrectly damped, amplified or redistributed.

If there are no restrictions on the relationship between time step size and the spatial increments in terms of stability, the scheme is unconditionally stable; otherwise it is conditionally stable [77].

2.6 Numerical Errors, Numerical Dispersion and Numerical Dissipation

Numerical errors in a scheme include the "amplitude error" and the "phase error". In finite difference methods, numerical dissipation and numerical dispersion are often used to quantify the errors. Mathematically, ω and β must be related. The relationship is known as the dispersion relation in EM theory, or more generally, the dispersion and dissipation relation [77]. Dissipation occurs when the amplitude of the plane wave decays or damps with time, and growth occurs when the amplitude increases with time. Dissipation causes smearing (i.e., sharp change is smoothed) and damping [77]. In physics, dispersion occurs when different wavelengths propagate at different speeds. In finite difference methods, dispersion also depends on the mesh shape, mesh density, time step size as well as direction of travel due to discretization, thus is called numerical dispersion.

When β is real, the numerical wave propagates at the speed of $u = \omega / \beta$, without growth or decay. If ω is a linear function of β, there is no dispersion in physics. To count for dissipation, $\omega = \omega(\beta)$ is expressed in a complex number $\omega_r + i\omega_i$. Thus $\omega_r(\beta)$ is the dispersion relation, and $\omega_i(\beta)$ is the dissipation relation [77].

All of the even-degree derivatives in the difference scheme result in dissipation or growth, and all of the odd-degree derivatives greater than one are dispersive [77]. With central differencing, all the even-degree terms cancel out. Thus most often FDTD methods using the central differencing are dispersive rather than dissipative.

2.7 Two-Level and Multilevel, Time Stepping, Explicit and Implicit Schemes

A two-level scheme involves only the quantities at time step n and n+1. If a scheme relates the quantities with three or more adjacent time steps, it is called "three-level" or "multilevel" method [49]. Yee's FDTD is "nominally" a multi-level method, but can be considered as two-level, since both the electric and the magnetic fields have only two time-levels involved though the time step indices are different [4]. Each ADI-FDTD sub-marching procedure is two-level, and Crank-Nicolson-based schemes [151] [156-157] [163] for the first-degree Maxwell's Equations are two-level methods, and are three-level methods for the second-degree wave equation [162].

If a scheme advances a full time step Δt with no intermediate time step, it is refereed as a "one-step" scheme; if a scheme advances successively through "sub-marching procedures", that is, advances a fractional time-step with one set of update equations, and then a further fractional time –step to complete Δt with another set of update equations; and each sub-marching procedure is associated with a specific time step, it is called "multi-time stepping" method [126]. Yee's FDTD is a one-step method, whereas ADI is two time-step method in which the intermediate time step is associated with $(n+1/2)\Delta t$.

An explicit scheme involves only one grid point at the current time level $t=(n+1)\Delta t$ and some grids at previous time steps [130]. Thus it permits step-by-step time advancement iteration directly [127]. An implicit scheme involves more than one grid point at the current time level. Such a formulation usually requires solving a sparse matrix at each sub-marching procedure or each time step. In general, both implicit scheme and explicit scheme can be unconditionally stable. Note that some implicit scheme is unstable, such as the first-order-approximation (Euler) scheme [49]. However,

for hyperbolic systems of PDEs, such as Maxwell's Equations, there are no explicit, unconditionally-stable, and consistent finite difference schemes [69].

2.8 Discretization of Boundary Conditions and Numerical Boundary Conditions

To perform a numerical solution, the boundary conditions have to be discretized. Improper implementation of the discretized BCs may ruin the accuracy of the numerical scheme, and may make the overall scheme unstable. Analysis shows that the discrete operator for the BCs must be close to $(q-1)^{th}$ order for q^{th}-order spatial operator [133].

The open region problem requires an extra, artificial "numerical" absorbing boundary condition (ABC) to "truncate" the computation into a "closed" problem, since a digital computer cannot store an infinite amount of data. Such an ABC is used to absorb ideally all outgoing waves at the outer boundaries from the region of interest, and to provide boundary values to complete the numerical scheme in the computational domain with $(i\Delta x, j\Delta y, k\Delta z)$ where $i = 1,2,...,M_x$, $j = 1,2,...,M_y$ and $k = 1,2,...,M_z$, where in total there are $(M_x - 1)(M_y - 1)(M_z - 1)$ cells. The outer boundary can be the $i = 1$ surface or the $i = M_x$ surface, etc. In FDTD methods, there are numerous papers on such ABCs [4] [6-7] [10]. This thesis will generally not address this issue except for a periodic BC for some implicit methods (see Chapter 6).

Note that the stability of an algorithm is not only related to the spatial increment and time step size, but also connected to schemes used to implement the boundary conditions [4] [54] [134]. References [68] [134] show that for an implicit non-iterative finite-difference scheme to be unconditionally stable, the numerical boundary conditions must be also implicit.

2.9 Numerical Experiments and Validation of A Scheme

Sometimes it is not possible to thoroughly analyze a numerical method to determine its properties of convergence, accuracy, stability, and consistency. Even if it is possible to analyze the properties, the theoretical result needs to be verified with numerical experiments to be sure that the theoretical assumptions that have been made are not deficient and that no mechanism has been neglected in the theoretical analysis. Therefore numerical experimentation must be performed to demonstrate the correctness. Such experiments must be carefully designed. One such experiment is the bench-mark problem test, in which either the analytical solution is known, or reliable measurement data or data from other well-established numerical methods have been published. However, such experiments may show good results only for the bench-mark problems, but poor results for many other real problems.

The phase velocity of wave propagation in a numerical method is approximate due to numerical dispersion [77]. An objective of the analysis of an FDTD method is to demonstrate how well it preserves the phase characteristics, or the true velocity of the wave in physics. Therefore, in this thesis, the numerical dispersion relations that are derived theoretically will be validated through numerical experiments by implementing the proposed scheme.

Another reason for numerical experiments is to test the numerical stability of the proposed methods by coding them and running the codes for many time steps. Every scheme proposed in this thesis has been tested and no violation to the stability analysis has been observed.

There are a variety of schemes in FDTD. Most schemes have one or more properties that make them desirable for specific types of problem. The preference of one to another depends on the problem and personal interest. Finally, it should be noted that the simulation results can only be as good as the model of the problem. Thus, a good model for a specific problem must reflect the real physics of the component or object. This thesis will devise methods for problems best suitable for the full-wave Maxwell's Equations.

2.10 Maxwell's Equations

Maxwell's Equations (MEs) [135] describe the temporal and spatial evolution of the electromagnetic waves, and have some special properties. First, MEs are a first-degree, symmetric and strongly hyperbolic system of PDEs [77]. Second, they are omni-direction wave equations, which is difficult to solve compared to the one-way wave equation $\partial U/\partial t + a\partial U/\partial x = 0$, where for $a > 0$ the wave travels along $+x$, and for $a < 0$ the wave travels along $-x$. Next, it is generally a mixed initial/boundary value problem. Thus the numerical solution for MEs requires special efforts and careful consideration. Though sometimes EM problems can be solved by approximating MEs with parabolic equations [12], in fact, the MEs are hyperbolic and will be solved directly for all the field components in this thesis. This is referred a "full wave" solution.

By using the MKS units, the time-dependent MEs can be given in differential or point form as Faradays' Law, Ampere's Law, Gauss' Law for the electric flux density and Gauss' Law for the magnetic flux density (Divergence Theorems) as follows [4] [135]

$$\partial \vec{B}/\partial t = -\nabla \times \vec{E} - \vec{M} \qquad (2\text{-}6a)$$

$$\partial \vec{D}/\partial t = \nabla \times \vec{H} - \vec{J} \qquad (2\text{-}6b)$$

$$\nabla \bullet \vec{D} = 0 \qquad \nabla \bullet \vec{B} = 0 \qquad (2\text{-}6c)$$

where it is assumed that there is no electric or magnetic current sources in the region of interest. In a linear (ϵ and μ are field-independent), isotropic (ϵ and μ are independent of direction), and non-dispersive material (ϵ and permeability μ are frequency-independent), the quantities can be related by the constitutive equations $\vec{D} = \varepsilon \vec{E}$, $\vec{B} = \mu \vec{H}$, $\vec{J} = \sigma \vec{E}$ and $\vec{M} = \sigma_m \vec{H}$, where σ is conductivity, and σ_m is the equivalent magnetic loss which is set to zero in this thesis. A lossless material has zero conductivity $\sigma = 0$, and a lossy material has larger-than zero conductivity $\sigma > 0$.

2.11 Yee's 2D Finite-Difference Time-Domain Method

In 1966, K.S.Yee published his milestone paper [14] for solving Maxwell's Equations by discretization in space and time with cells. The 2D Yee mesh is shown in Fig.2-1.

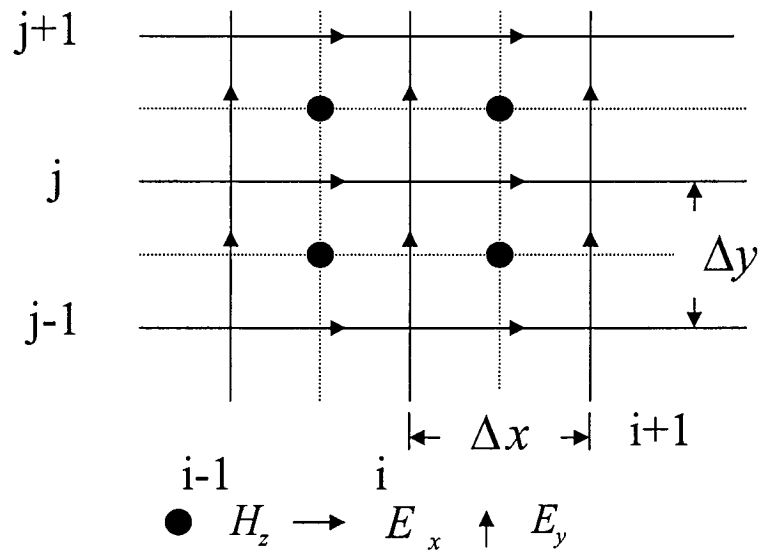

Fig. 2-1 Regular 2D Yee mesh and the locations of the field components.

The Yee mesh is a regular, orthogonal, dual-staggered grid consisting of a primary grid (solid line) and a secondary grid (dashed line). For the 2D TE_z wave, the field

components are space-staggered (E offsets H), and uncollocated (E_x is separated from E_y). The magnetic field H_z is sampled at the intersection of the dashed lines, and the electric field component E_x at the intersection of the horizontal solid lines and the vertical dash lines, but E_y is sampled in the intersection of the vertical sold lines and the horizontal dash lines. Those intersections are the "nodes". The time levels in the time-derivative terms are interleaved by the "leapfrog" scheme over the time levels in the space-derivative term.

In finite-difference time-domain methods, the vector components of the curl operators in Cartesian coordinates can be expressed with coupled scalar equations, and they are the basis of the FDTD numerical algorithm [4]. The divergence Theorems are automatically satisfied [6], which are centered at the intersections of two solid lines (for the electric field) and two dash lines (for the magnetic field for TM_z wave). Thus Yee's difference or the update equations in linear, isotropic, non-dispersive and lossless media are

$$H_z^{n+1/2}(i+1/2,j+1/2)=H_z^{n-1/2}(i+1/2,j+1/2)+ a_2\frac{E_x^n(i+1/2,j+1)-E_x^n(i+1/2,j)}{\Delta y}-a_2\frac{E_y^n(i+1,j+1/2)-E_y^n(i,j+1/2)}{\Delta x} \tag{2-7a}$$

$$E_x^{n+1}(i+1/2,j)=E_x^n(i+1/2,j)+a_1\frac{H_z^{n+1/2}(i+1/2,j+1/2)-H_z^{n+1/2}(i+1/2,j-1/2)}{\Delta y} \tag{2-7b}$$

$$E_y^{n+1}(i,j+1/2)=E_y^n(i,j+1/2)-a_1\frac{H_z^{n+1/2}(i+1/2,j+1/2)-H_z^{n+1/2}(i-1/2,j+1/2)}{\Delta x} \tag{2-7c}$$

where $a_1=\Delta t/\varepsilon$, $a_2=\Delta t/\mu$. A 2D Yee cell contains $E_x(i+1/2,j)$, $E_y(i,j+1/2)$ and $H_z(i+1/2,j+1/2)$ as can be seen in Fig. 2-1. The staggered arrangement of the electric and magnetic field components is consistent with the boundary conditions where the tangential E and H at the grid surfaces are continuous, allowing the electric and magnetic

fields to be alternately computed in a leapfrog manner so that the coupled field components are "decoupled" in time. Yee's FDTD is explicit, one time-step method, thus is relatively fast, easy to implement, and requires moderate computer memory. Note that Yee's mesh is "ordered", "structured".

2.12 Stability Analysis and Numerical Dispersion of 2D Yee's FDTD.

Using the Fourier analysis method with Eqn. (2-5), the update equations (2-7) can be simplified to

$$QW^n = 0 \tag{2-8a}$$

$$Q = \begin{pmatrix} 1-\xi & 0 & -J2a_1 \sin(\beta_y \Delta y/2)/\Delta y \\ 0 & 1-\xi & J2a_1 \sin(\beta_x \Delta x/2)/\Delta x \\ -J2a_2 \sin(\beta_y \Delta y/2)/\Delta y & J2a_2 \sin(\beta_x \Delta x/2)/\Delta x) & 1-\xi \end{pmatrix} \tag{2-8b}$$

$$W^n = \begin{pmatrix} E_x^n & E_y^n & H_z^n \end{pmatrix}^T \tag{2-8c}$$

The amplification polynomial is the determinant of the amplification matrix Q, which is

$$P(\xi) = (1-\xi)(1 + 2(2r_x^2 + 2r_y^2 - 1)\xi + \xi^2) \tag{2-9}$$

where $r_x = c\Delta t \sin(\beta_x \Delta x/2)/\Delta x$, $r_y = c\Delta t \sin(\beta_y \Delta y/2)/\Delta y$. The amplification factor ξ can be found from the roots of $P(\xi)$. The first root is $\xi = 1$, which is a stationary, non-propagating solution, and the non-stationary solution is

$$\xi = \exp\left(\pm J \tan^{-1} \frac{2\sqrt{(r_x^2 + r_y^2)(1 - r_x^2 - r_y^2)}}{1 - (r_x^2 + r_y^2)} \right) \tag{2-10}$$

which is the propagating mode solution. Eqn. (2-10) is valid with the condition that $1 - (r_x^2 + r_y^2) \geq 0$ is satisfied. The maximum of the sine is unity, that is, $\max\{r_x^2, r_y^2\} = \{c\Delta t/\Delta x, c\Delta t/\Delta y\}$, thus the condition can be simplified as

$$\Delta t \le \frac{1}{c\sqrt{1/\Delta x^2 + 1/\Delta y^2}} \tag{2-11}$$

This is the stability condition, usually referred as the Courant-Friedrich-Levy (CFL) limit or Courant limit [49] [69] [129], and is a general requirement for hyperbolic explicit difference equations. If a scheme satisfies the CFL condition, then it is "convergent" [77].

The Courant number (CFLN) is defined as $s = c\Delta t / \Delta_{\max}$ in general in this thesis where $\Delta_{\max} = \max\{\Delta x, \Delta y, \Delta z\}$. For a homogenous or uniform mesh where all spatial increments are identical, that is, $\Delta x = \Delta y = \Delta z$, the Courant condition requires $s \le 1/\sqrt{d}$ for 1D, 2D and 3D where $d = 1,2,3$ is the dimension. Another parameter called "relative Courant number" (RCFLN) [165] is the product of the CFLN and the aspect ratio $R = \sqrt{(\Delta_{\max}/\Delta x)^2 + (\Delta_{\max}/\Delta y)^2 + (\Delta_{\max}/\Delta z)^2}$, which indicates that how big the time step size is compared to the Courant limit. In Yee's FDTD, the Courant limit insures the time step size is much less than the limit of the Nyquist criterion.

Since the amplification factor in Eqn. (2-10) has a unity magnitude, either substituting $\xi = \exp(\omega\Delta t/2)$ into the update equations, or comparing $\xi = \exp(\omega\Delta t/2)$ to Eqn. (2-10) can obtain the numerical dispersion relation. The numerical dispersion relation can be derived as [4]

$$\frac{\sin^2(\omega\Delta t/2)}{(c\Delta t)^2} = \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2} + \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} \tag{2-12}$$

By use of simple root-finding algorithm, the numerical dispersion can be evaluated. Numerical anisotropy occurs when a plane wave has different numerical velocity of propagation in different directions [4].

2.13 Source Excitation

A complete FDTD implementation must include a computationally-efficient source excitation. In order to simulate the propagation of EM waves, the source in the FDTD lattice must be coupled into the FDTD grid properly. Depending on different properties and criteria, many terminologies [4] [136-138] are used to describe an FDTD source. This thesis uses a pulsed, monochromatic, embedded (internal), "hard" field source.

FDTD is devised to determine the transient EM fields. It can be regarded as a "causal" system with a starting point $t = 0$ [139]. If the driving temporal waveform is not properly chosen, static fields may be included in the FDTD results [137]. To avoid the induced, unintended dc-offset static field, a zero mean-value function is usually used, such as the bi-polar [138] (also called double or differential) Gaussian pulses. The source parameters should be chosen carefully to cover the desired frequency band, and at the same time avoid unwanted higher frequencies and overshoot [139] [158] due to sudden turn-on or turn-off. In this thesis, the following driving function is used

$$H_s = \begin{cases} e^{-(\frac{n-n_0}{w_0})^2} \sin(\omega n \Delta t) & n \le n_0 \\ \sin(\omega n \Delta t) & n > n_0 \end{cases} \tag{2-13}$$

where n_0 and w_0 are parameters to control the "turn-on" transient of the excitation and $w_0 = n_0 / 4$ [6], and ω is the designate angular frequency of the signal. Such a waveform may be called "quasi-monochromatic" since it has a non-monochromatic part only during the turn-on transient.

2.14 Numerical Experiment Design and Extraction of the Numerical Velocity

To validate theoretical numerical dispersion relations, their predictions must be compared with numerical experiments. To extract the numerical velocity from the FDTD

simulation, usually the data has to be transformed to the frequency domain [8]. This thesis directly uses the time domain data to obtain the numerical velocity with the "matching method" [158].

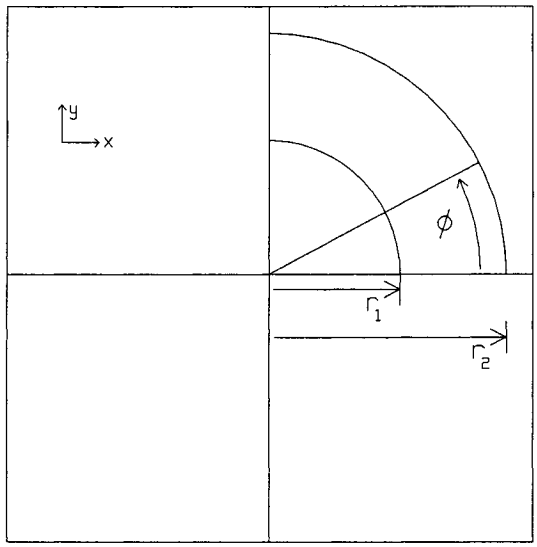

Fig. 2-2 Observers in the computational domain.

2.14.1 The matching method

The matching method requires two sets of data: one set is the field values recorded in time at a "close" observer located at r_1, and the other set is those obtained at a "far" observer along a direction of travel at r_2, as shown in Fig.2-2 in 2D case. Since the distance between the pair of observers is known, the numerical velocity u can be calculated with

$$u = \frac{r_2 - r_1}{\delta t} \tag{2-14}$$

where δt is the time delay as the wave travels from the close observer at r_1 to the far observer at r_2. Then the numerical dispersion is evaluated as u/c [4].

The "matching method" was devised to accurately find the time delay δt. It first normalizes the amplitude at the far observer to remove the distance dependence. Then the normalized field h_{r2} at the far observer is "matched" to h_{r1} at the close observer by dragging the far data along the time axis until it coincides with the data at the near observer. The required time shift is the time delay δt. "Coincidence" is defined to be minimization of the error, given by $e = \sum_{m=1}^{n} \left(h_{r1}(m\Delta t) - h_{r2}(m\Delta t - \delta t)\right)^2$. The "dragging" operation is performed numerically by evaluating the error as a function of δt; the value of δt that minimizes the error is the time delay.

To implement the matching method efficiently, a shift-and-compare algorithm is used. First a "coarse" change of the value of δt is used to achieve an approximate match, then a "fine" change in δt is used to refine the match by minimizing the error. Thus the time delay between the pair of observers is found to the accuracy of the fine time step size. More detail can be found in [158].

Each 1D and 2D FDTD method proposed in this thesis has been coded and run. The codes include subroutines to locate two sets of 91 observers in 2D, one degree of arc apart, and 2 observers in 1D, and to interpolate the field value at the observer using those at adjacent grid nodes. The numerical velocity is found from the numerical experiments by running the code of the matching method. In order to have acceptable accuracy in finding the numerical velocity, the experiment must be carefully designed.

2.14.2 Numerical experiment design

The Yee mesh can be regarded as a time-response system, and it deals with transient signals during the turn-on period. Such a system is low-pass, and frequencies higher than the Nyquist criterion $f_{Nyquist} = (2\Delta t)^{-1}$ (the cut-off frequency) cannot be propagated. This

Nyquist frequency corresponds to a minimum mesh density $N_{\min} = 2$ cells per wavelength (CPW). In practice, the mesh density is often chosen to be around 10 CPWs for Yee's FDTD [3], and much denser to resolve the small geometrical features for ADI and CN- based unconditionally-stable methods. However, numerical experiments show that for whatever the mesh density, there is "overshoot" (shown in Figs. 2 and 3 in [158]), meaning that the field value at the far observer is not an exact duplicate of the field value at the close observer, that is $h_{r2}(t-\delta t, r_2) \neq h_{r1}(t, r_1)$. Because the FDTD mesh is a discrete system with a finite response, the overshoot is inherent, and cannot be avoided.

The Gaussian function in (2-13) is used to reduce the overshoot to an "acceptable" level by using a "slow" turn-on transient. The value of parameter n_0 depends on the mesh density and the Courant number. The numerical experiment design includes choosing a proper value for n_0, the size of the computational space M_x and M_y for a given mesh density of an FDTD method to avoid reflection from the boundary.

For Yee's FDTD, since the Courant number is smaller than unity, the commonly-acceptable value for n_0 is 32, which has a source value of -70dB [6]. For explicit methods proposed in this thesis, n_0 is chosen to be 32. However, for larger Courant numbers used in unconditionally-stable FDTD methods, experiments show that large overshoot appears. For a Courant number 10 and mesh density 100 CPWs, $n_0 = 320$ works well.

To evaluate the numerical dispersion with the Courant number s and mesh density N CPWs, the time step size is

$$\Delta t = \frac{s}{N}\frac{2\pi}{\omega} \tag{2-15}$$

Each sinusoidal cycle has N/s data points. Next, the total number of data points will be calculated for a specific experiment. When the far observer is located five wavelengths away from the source, $5N/s$ time steps are needed for the wave to reach the far observer in the physical world. Because the source requires 320 time steps to reach steady state, the first few cycles with less than one-tenth of the amplitude value must be discarded for velocity extraction. This is equivalent to about 120 time steps. For the matching method to be effective, about 8 to 10 cycles of the sine wave are required at the far observer, because the source is still in the transient state. The thesis uses the largest Courant number of $s = 10$, and the finest mesh density of $N = 100$ CPWs for numerical experiments, The total number of time steps is $5 \times 100/10 + 120 + 8 * 100/10 = 250$, which is still less than the turn-on transient of 320 time steps. Since the outer boundary is PEC in this thesis, reflection from the boundary would ruin the data. The cell space size is chosen so that in 250 time steps, the wave does not reach the outer boundary. Thus $250 \times 100/10 = 2500$ cells from the generator to the boundary are sufficient, and with the source at the center, a computational space of 5000 by 5000 cells is used. Note that the FDTD algorithm does not have to run for enough time steps to reach the sinusoidal steady state.

2.14.3 Accuracy of the matching method

Numerical experiment results show that the matching method is very accurate to extract the velocity directly from the FDTD data, even from the turn-on transient data [158]. For Yee's FDTD, the mesh density is 10 CPWs for 300 MHz signal; the time step size is the Courant limit $s = 0.707$, at locations $r_1 = 15\,\text{m}$ and $r_2 = 30\,\text{m}$ in a 1000 by 1000 cells space, the numerical dispersion along the axis is calculated with the matching

method to be 0.991472. The exact dispersion formula Eqn. (2-12) gives 0.991485. The absolute error between the theoretical predication and the numerical results is 1.3×10^{-5}. When substituting the velocity from numerical experiment into Eqn. (2-12), the residual error 1.6×10^{-4}.

For ADI-FDTD of Courant number 10, mesh density 100 CPW in a 5000 by 5000 cells space, the numerical dispersion values obtained using the matching method are: 0.967192 along x-axis, 0.978763 along diagonal 45°, and 0.967041 along y-axis, respectively [158]. The theoretical values are 0.966713, 0.979159 and 0.966713, respectively. The absolute errors between the theoretical predication and the numerical results are 4.8×10^{-4} along x-axis, 4.0×10^{-4} along diagonal 45°, and 3.3×10^{-4} along y-axis, respectively. Substituting these numerical dispersion values into the numerical dispersion relation, the residual errors are 1×10^{-6}, 9×10^{-7}, and 7×10^{-7}, respectively. For Courant number 1, the maximum absolute error is 2.1×10^{-4}, and the residual error is 4.1×10^{-7}. This is a sufficient accuracy to affirm that the numerical dispersion relation is correct. Therefore through the accurate validation of the numerical dispersion relation, the proposed methods are also validated.

2.15 Summary

This chapter introduces some basic concepts in finite difference methods and FDTD. The key issue in analyzing an FDTD scheme is the stability. According to the Lax theorem, proving the stability of a difference scheme ensures the consistency and convergence. Thus this thesis will only prove the stability of a proposed scheme by the use of the Fourier method, which is a sufficient and necessary tool in analyzing stability of an FDTD scheme.

Numerical schemes can be explicit or implicit, conditionally or unconditionally stable. For hyperbolic system Maxwell's Equations to be unconditionally stable, the scheme must be implicit. If the scheme is implicit, the boundary condition must also be implicit, and can be one-order less accurate then the scheme.

For FDTD method, the central differencing formulas are used to avoid dissipation. The order of accuracy of a scheme is the lowest order of the truncation error. To evaluate a scheme, the numerical dispersion relation will be analyzed and validated with numerical experimentation. The matching method is quite accurate for the extraction of the numerical velocity from numerical experiments.

Yee's FDTD solves the Maxwell's Curl equations explicitly, using a space-staggered grid for all the field components, and interleaved time-stepping between electric and magnetic field components. However, Yee's FDTD, like other explicit time domain methods, suffers from numerical dispersion and anisotropy errors, and most importantly, it has the Courant time-step-size limit.

In Chapter 3, some explicit methods that can eliminate or reduce numerical dispersion and anisotropy are proposed. In Chapters 4, 5, and 6, some unconditionally-stable FDTD methods will be introduced to overcome the Courant limit.

Chapter 3 New Explicit Methods for Suppression and Elimination of Numerical Dispersion and Anisotropy

Current explicit FDTD (2, 2) methods are usually efficient for moderate electrical sizes, on the order of 20 or less [33]. For larger problems, higher-order methods are often used which have smaller accumulated phase error than 2^{nd} order methods. However, higher-order methods usually need higher-order boundaries and material interfaces, which complicate the computation. This chapter proposes optimized methods with better accuracy by using the 2^{nd}-order accurate finite differencing, and the boundaries are easily treated using current techniques for 2^{nd} order methods.

3.1 Introduction

Yee's FDTD suffers from numerical dispersion and the anisotropy giving rise to a direction-dependent phase error. Since the numerical velocity of propagation is dependent on the mesh size, time step size, and the direction of travel [4], the anisotropic numerical wave complicates the near-to-far transformation [140]. Of course, a finer mesh can be used to reduce the numerical error, but at the expense of longer CPU running time and increase of computer memory requirement, both because smaller cells implies many more cells to fill a given volume of space, and because smaller cells require a smaller time step size for stability. Consequently, many researchers have tried to develop new FDTD algorithms that have less dispersion error than Yee's for a given mesh density, and a larger time step limit for stability. A short literature survey is as follows.

Reference [26] first presented a high-order (2, 4) method based on Taylor series analysis. Other higher-order methods are summarized in Chapter 1. Reference [39]

proposed a method for increasing the velocity without increasing the processing time or memory requirement, resulting in zero phase error in an average sense at one frequency. Reference [36] suggested a method using artificial anisotropy by changing the dielectric constants in the algorithm. By the combination of two different Laplacian difference operators with a non-standard finite-difference formula, References [40-41] constructed a new difference operator with up to 6^{th} order truncation error. To reduce the numerical dispersion in a specific direction, References [45-48] presented an angle-optimized scheme and a dispersion-preserving algorithm with digital filtering. With the improvement of numerical dispersion, the numerical anisotropy decreases but does not vanish. References [33-35] compared the dispersion properties of several low-dispersion FDTD methods.

This thesis proposes the following methods: a weighted method with optimization of the higher-order (2, 4) scheme (O24) [159] and a "neighborhood averaging" (NA) method; these two methods can eliminate numerical dispersion along some specific directions, or even remove anisotropy completely at one frequency, but with residual numerical dispersion, for a uniform mesh (square cells). By combining the two methods and optimization, a novel scheme termed the "neighborhood-averaging 24-stencil" (NA24) method can eliminate anisotropy and dispersion at the same time at one frequency, and reduce greatly the dispersion and anisotropy for other frequencies. The O24 method has the same time step size limit as the (2, 4) scheme, which is smaller (6/7) than the Courant limit, NA method has a larger than the Courant limit, and the NA24 method has the same Courant limit as Yee's FDTD.

To compare numerical dispersion, the numerical dispersion or the relative velocity $u_r = u/c$ is often used where u is the numerical velocity as a function of the direction of travel. It gives information of the numerical dispersion error and the anisotropy at one frequency or mesh density. The second evaluation is the maximum numerical dispersion error as a function of mesh density, which gives an idea how accurate a method is over a broad bandwidth. To evaluate the uniformity of the numerical velocity, anisotropy defined as

$$A_u = \frac{u_d - u_a}{\min\{u_d, u_a\}} \times 100\% \tag{3-1}$$

is also used, where u_d and u_a are numerical velocities along the diagonal and the axis. Another important evaluation is the accumulated phase error, which is defined as

$$p_{err} = 360^\circ \left(1 - \frac{c}{u}\right) \frac{\Delta x}{\lambda} \tag{3-2}$$

It is the phase error per cell of distance traveled. The total phase error along one direction of travel is simply the product of the accumulated phase error per cell with the electrical distance the wave travels. Because of anisotropy, the maximum accumulated phase error versus mesh density is often used.

Traditionally, higher-order (higher than 2nd order) schemes are evaluated based on the Taylor's Series. In this Chapter, we do not pursue higher-than-2nd-order accuracy based on the Taylor Series. Instead, the proposed methods are based on the optimization of the numerical dispersion, which leads to smaller numerical error in some sense. For simplicity, a 2D TE_z wave in a linear, isotropic, non-dispersive medium is assumed.

3.2 Optimized (2, 4) Stencil Scheme

Based on the Taylor Series analysis [4] [26] [28] the first spatial derivatives in the (2, 4) scheme are approximated by the use of the conventional Yee's elements plus "one-cell-away" elements. Both elements use a central finite-difference formula to eliminate third-order and higher odd-order terms. Fourth-order accuracy is obtained by the cancellation of the two second-order error terms from the Yee's elements and the "one-cell-away" elements. The resulting numerical anisotropy is smaller than that of Yee's method, but with numerical velocity higher than speed of light. Based on the same stencil, this Chapter discusses the optimization of the numerical dispersion. To achieve this, a weight parameter w_{24} is introduced to optimize the relative contributions of the Yee's elements and the "one-cell-away" elements. The new update equations are [159]

$$H_z^{n+1/2}(i+1/2,j+1/2)=H_z^{n-1/2}(i+1/2,j+1/2)+w_{24}a_2\{[E_x^n(i+1/2,j+1)-E_x^n(i+1/2,j)]/\Delta y -[E_y^n(i+1,j+1/2)-E_y^n(i,j+1/2)]/\Delta x\}+(1-w_{24})a_2\{ [E_x^n(i+1/2,j+2)-E_x^n(i+1/2,j-1)]/3\Delta y-[E_y^n(i+2,j+1/2)-E_y^n(i-1,j+1/2)]/3\Delta x\} \quad (3\text{-}3a)$$

$$E_x^{n+1}(i+1/2,j)=E_x^n(i+1/2,j)+w_{24}a_1\{[H_z^{n+1/2}(i+1/2,j+1/2)-H_z^{n+1/2}(i+1/2,j-1/2)]/\Delta y\}+ (1-w_{24})a_1\{[H_z^{n+1/2}(i+1/2,j+3/2)-H_z^{n+1/2}(i+1/2,j-3/2)]/3\Delta y\} \quad (3\text{-}3b)$$

$$E_y^{n+1}(i,j+1/2)=E_y^n(i,j+1/2)-w_{24}a_1\{[H_z^{n+1/2}(i+1/2,j+1/2)-H_z^{n+1/2}(i-1/2,j+1/2)]/\Delta x\}+ (1-w_{24})\{[H_z^{n+1/2}(i+3/2,j+1/2)-H_z^{n+1/2}(i-3/2,j+1/2)]/3\Delta x\} \quad (3\text{-}3c)$$

Because Eqn. (3-3) uses the same stencil as the standard (2,4) stencil, this method is termed the "optimized (2,4)" (O24) method. Using the Fourier analysis method, the amplification factor for the O24 formulation (3-3) can be obtained as

$$\xi = 1-2\left(r_x^2(w_{24}+(1-w_{24})p_{x24})^2+r_y^2(w_{24}+(1-w_{24})p_{y24})^2\right) \pm J\sqrt{1-\left(1-2\left(r_x^2(w_{24}+(1-w_{24})p_{x24})^2+r_y^2(w_{24}+(1-w_{24})p_{y24})^2\right)\right)^2} = \exp\left(\pm J\tan^{-1}\frac{\sqrt{1-\left(1-2\left(r_x^2(w_{24}+(1-w_{24})p_{x24})^2+r_y^2(w_{24}+(1-w_{24})p_{y24})^2\right)\right)^2}}{1-2\left(r_x^2(w_{24}+(1-w_{24})p_{x24})^2+r_y^2(w_{24}+(1-w_{24})p_{y24})^2\right)}\right) \quad (3\text{-}4a)$$

where

$$p_{x24} = 1 - 4\sin^2(\beta_x \Delta x/2)/3 \tag{3-4b}$$

$$p_{y24} = 1 - 4\sin^2(\beta_y \Delta y/2)/3 \tag{3-4c}$$

With some manipulation, the numerical dispersion equation can be obtained as

$$\left(\frac{\sin(\omega\Delta t/2)}{c\Delta t}\right)^2 = \left(w_{24}\frac{\sin(\beta_x\Delta x/2)}{\Delta x} + (1-w_{24})\frac{\sin(3\beta_x\Delta x/2)}{3\Delta x}\right)^2 + \left(w_{24}\frac{\sin(\beta_y\Delta y/2)}{\Delta y} + (1-w_{24})\frac{\sin(3\beta_y\Delta y/2)}{3\Delta y}\right)^2 \tag{3-5}$$

Since the numerical dispersion depends on the weight parameter, the dispersion behavior can be optimized in various ways. In practice, some problems, such as the laser cavity and other examples indicated in [2] [45-46], require an FDTD method that has less numerical error within an angular sector of space, because most waves travel within that sector. Therefore in next sub-section, a general formula for arbitrary-angle optimization will be given with three specific applications.

3.2.1 Arbitrary-angle optimization

The formulation of Eqn. (3-3) allows the elimination of the numerical dispersion at any specific angle of interest ϕ_0. The weight parameter can be found from

$$\left(\frac{\sin(\omega\Delta t/2)}{c\Delta t}\right)^2 = \left(w_{24}\frac{\sin(\beta_{x\phi_0}\Delta x/2)}{\Delta x} + (1-w_{24})\frac{\sin(3\beta_{x\phi_0}\Delta x/2)}{3\Delta x}\right)^2 + \left(w_{24}\frac{\sin(\beta_{y\phi_0}\Delta y/2)}{\Delta y} + (1-w_{24})\frac{\sin(3\beta_{y\phi_0}\Delta y/2)}{3\Delta y}\right)^2 \tag{3-6a}$$

where $\beta_{x\phi_0} = \beta_0\cos(\phi_0)$, and $\beta_{y\phi_0} = \beta_0\sin(\phi_0)$. The solution for w is

$$w_{24} = \frac{A + \sqrt{B\sin^2(\omega\Delta t/2) - C}}{B} \tag{3-6b}$$

$$A = r_{x\phi_0}^2 p_{x\phi_0}(p_{x\phi_0} - 1) + r_{y\phi_0}^2 p_{y\phi_0}(p_{y\phi_0} - 1)$$

$$B = r_{x\phi_0}^2 (p_{x\phi_0} - 1)^2 + r_{y\phi_0}^2 (p_{y\phi_0} - 1)^2 \qquad (3\text{-}6c)$$
$$C = r_{x\phi_0}^2 r_{y\phi_0}^2 (p_{x\phi_0} - p_{y\phi_0})^2$$

To demonstrate the feasibility of the method, the "axis-optimized method" (AOM), the "diagonally optimized method" (DOM) and the "minimum-average-dispersion" method (MADM) will be discussed next.

A. Axes-optimized method

The axes-optimized method has no numerical dispersion along the axes for any time step size within the stability limit at a designated frequency. To eliminate the numerical dispersion error along the axes where $\phi_0 = 0^0$ or $\phi_0 = 90^0$, for a square mesh ($\Delta x = \Delta y$), the following optimal value of the weight parameter [159] can be used

$$w_{AOM} = \frac{\left(\frac{\sin(\omega\Delta t/2)}{c\Delta t} - \frac{\sin(3\beta_0\Delta x/2)}{3\Delta x}\right)}{\left(\frac{\sin(\beta_0\Delta x/2)}{\Delta x} - \frac{\sin(3\beta_0\Delta x/2)}{3\Delta x}\right)} \qquad (3\text{-}7)$$

where $\beta_0 = 2\pi/\lambda_0$ is the physical phase constant, and λ_0 is the designated wavelength for zero numerical dispersion along the axes. To validate the formulation Eqs. (3-3), (3-5) and (3-7), numerical experiments have been performed. In this Chapter, a grid of 2000 by 2000 cells is used, mesh densities of 10, 20 and 30 cells per wavelength (CPW) are tested. The numerical velocity is extracted from the experiment data with the matching method introduced in Chapter 2. The numerical dispersion of the theoretical predication agrees very well with that from the numerical results (see Fig. 2 in [159]). Fig. 3-1 graphs the maximum percent numerical error (MPNE) defined as

$$\left|1 - u/c\right|_{\max} \times 100 \qquad (3\text{-}8)$$

along the axis and at $\phi = \pm 10°$, at the Courant number $s = 0.50 s_c$ and $s = 0.848 s_c$, where $s_c = 1/\sqrt{2}$ is the Courant limit in the 2D case. It can be seen that the numerical dispersion is much smaller than the minimum dispersion for the standard (2, 4) method (the maximum dispersion error of the (2, 4) method is too large to be shown).

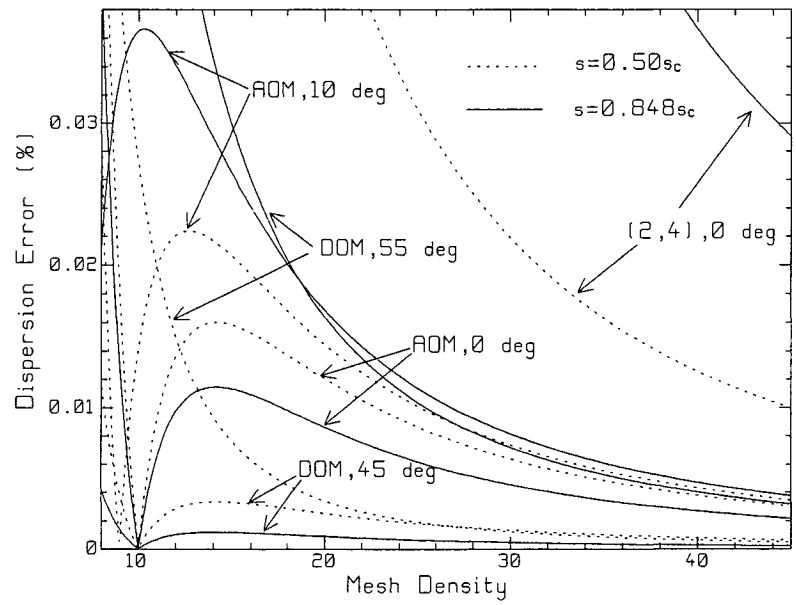

Fig. 3-1 The maximum percent numerical dispersion error for AOM along 0° and 10°, DOM along 45° and 55°, optimized at 10 CPW and the minimum dispersion error of the (2, 4) scheme along the axes.

B. Diagonally-optimized method

The "diagonally optimized method" (DOM) has no numerical dispersion error along $\phi_0 = 45°$ for a square mesh. The optimal value of the weight parameter w_{DOM} is [159]

$$w_{DOM} = \frac{\left(\frac{\sin(\omega \Delta t/2)}{\sqrt{2} c \Delta t} - \frac{\sin(3\sqrt{2}\beta_0 \Delta x/4)}{3\Delta x} \right)}{\left(\frac{\sin(\beta_0 \Delta x/2)}{\Delta x} - \frac{\sin(3\sqrt{2}\beta_0 \Delta x/4)}{3\Delta x} \right)} \tag{3-9}$$

Numerical experiments show good agreement with the theory (Fig.4 in [159]). Fig. 3-1 shows the maximum percentage dispersion error optimized at 10 CPW. It can be seen that the MPNE for DOM is much smaller than that of (2, 4), and smaller than that of AOM in a broad bandwidth. For example, at $s = 0.5s_c$, the maximum dispersion error for DOM occurs at about 14 CPW, which is 0.00336% along the diagonal, and 0.00937% at $\phi = 55°$, about 25.7 and 9.2 times smaller than the minimum dispersion error of the (2,4) scheme, respectively, at the same mesh density.

C. The minimum-average-dispersion method

The "minimum-average-dispersion" method eliminates the numerical dispersion error along $\phi = 22.5^{\circ}$ and $\phi = 67.5°$ using square mesh cells. For a time step size $s = 0.9324 = 1.3186s_c$, the optimal value is 1.054321, with Courant number limit of 0.932417 from Eqn. (3-6a). When optimized at 10 CPW, the largest numerical dispersion error and the averaged dispersion error are less than 0.0025 and 0.003545 compared to 0.00589 and 0.568 for the standard (2,4) method at 10 CPW. Numerical experiments show very good agreement with the theory. This method has twice smaller the numerical dispersion error than the (2,4) method with a larger time step size limit. In addition, it has eight zero-dispersion sectors angles at 22.5°, 67.5°, 115.5°, 157.5°, 202.5°, 247.5°, 292.5° and 337.5°, compared to four such angles in AOM and DOM.

3.2.2 Isotropic optimization method

The above discussion has shown that the weight parameter w_{24} can be chosen to make the numerical phase constant exactly equal to the theoretical value in one direction of travel. The values of the weight parameter Eqs. (3-7) and (3-9) depend on the mesh density and the time step size. If the AOM and DOM methods are combined together, the

numerical velocity can be made independent of the direction of travel. This method is termed as "isotropic optimization method" (IOM). With some manipulation, the "isotropic" optimal value of the weight parameter for IOM is derived as [159]

$$w_{24i} = \frac{\left(\sqrt{2}\frac{\sin(3\sqrt{2}\beta_0\Delta x/4)}{3\Delta x} - \frac{\sin(3\beta_{0a}\Delta x/2)}{3\Delta x}\right)}{\left(\frac{\sin(\beta_0\Delta x/2)}{\Delta x} - \frac{\sin(3\beta_0\Delta x/2)}{3\Delta x}\right) - \sqrt{2}\left(\frac{\sin(\sqrt{2}\beta_0\Delta x/4)}{\Delta x} - \frac{\sin(3\sqrt{2}\beta_0\Delta x/4)}{3\Delta x}\right)} \quad (3\text{-}10)$$

It can be seen that optimal parameter w_{24i} is only a function of mesh density at the optimized wavelength. Fig. 3-2 graphs the numerical dispersion of the theoretical prediction (solid line) and numerical results (small circles), optimized at 10 CPW. For comparison, the numerical dispersion of the (2, 4) method is shown as dashed lines.

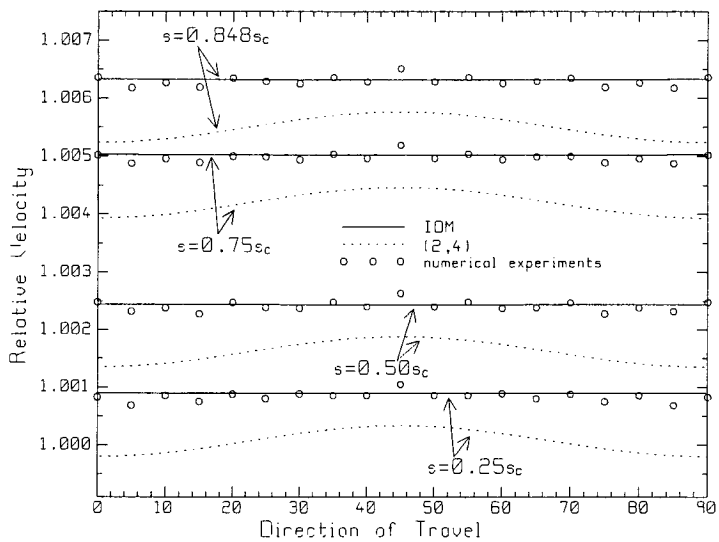

Fig. 3-2 Numerical dispersion for the isotropic optimization method (IOM) optimized at 10 CPW and for the (2,4) method.

The small difference of the numerical results and theory is in the accuracy limit of the matching method, which is in the order of 10^{-4} due to the use of fewer time steps to speed up the calculation. The maximum anisotropy is 1.4×10^{-4} at 14 CPW when

optimized at 10 CPW. It is noticed that a very recent paper reported a similar method [43], but optimized with an approximation of the numerical dispersion relation.

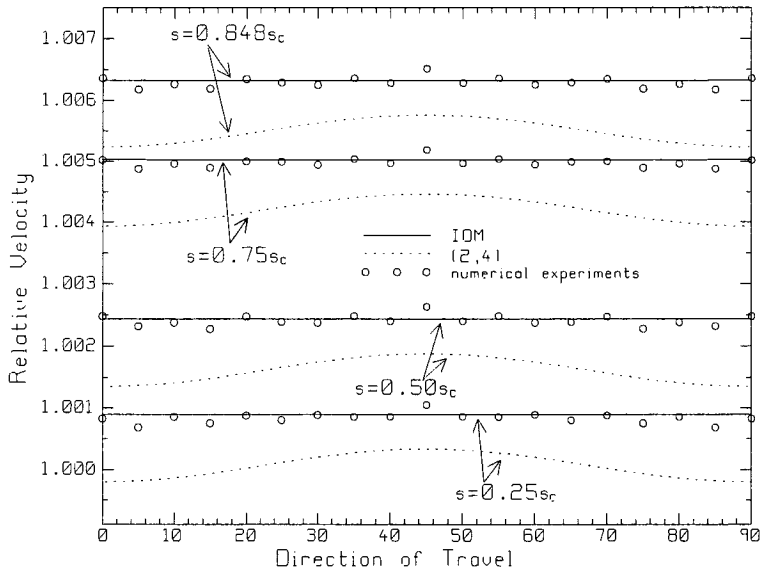

Fig. 3-2 Numerical dispersion for the isotropic optimization method (IOM) optimized at 10 CPW and for the (2,4) method.

3.2.3 A high-accuracy-method with fixed-weight

The (2, 4), AOM, DOM and IOM methods have larger dispersion error as the time step size increases, which is contrary to the Yee's method. In addition, their numerical phase velocity is generally larger than the physical speed. Because these methods are based on the weighted contributions of Yee's elements and the "one-cell-away" elements, the larger numerical velocity indicates that the methods are "over-weighted". When the weight parameter becomes smaller, the numerical velocity becomes smaller, and approaches to the physical speed. In this section we will find a constant value of the weight parameter that is "optimal" in terms of numerical dispersion error.

Expanding the sine terms in Eqn. (3-10) with the Taylor' series up to the 3rd order, a constant value of the weight parameter is obtained as $w_{fw} = 27/26$. Fig. 3-3 shows the

numerical dispersion from theory (solid line) and from the numerical experiments (small circles) with very good agreement. Comparing to the (2, 4) method and IOM method, this method has higher numerical accuracy in terms of the "averaged-accumulated phase error" [159], thus is termed "high-accuracy-method with fixed weight" (HAM-FW).

3.2.4 High accuracy method by search

The optimal value of the weight parameter can also be found by a search algorithm once the optimization criterion is set. For a range of weight parameter from $w_{24} = 1.0000$ to 1.4000 with a increment of 0.0001, and for Courant numbers from $s = 0.8$ to 1.2 with a increment of 0.0001, a simple search algorithm was used to find values of w where the error $\left\| 1 - u(\phi = 0°)/c \right| - \left| 1 - u(\phi = 45°)/c \right\|$ is less than 10^{-5} and $1 - u(\phi = 0°)/c > 0$.

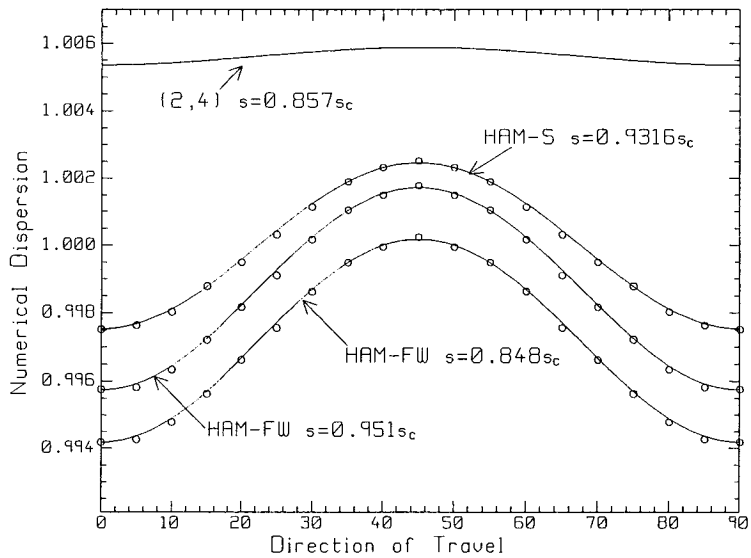

Fig. 3-3 Numerical dispersion of the high-accuracy-method with fixed-weight HAM-FW with $w_{fw} = 27/26$ and the high-accuracy-method by searching HAM-S.

The search finds (w_{24}, s) pairs which meet the error criterion. However, the time step size $\Delta t = s\Delta t_c$ should not be larger than the upper bound as $s_{\max} = 3/(4w_{24} - 1)$ [159].

This eliminates many (w_{24}, s) pairs, leaving useful pairs with w in the range from 1.0546 to 1.0743, which correspond to a time step size limit larger than 0.9 times the Courant limit of Yee's FDTD. For the pair ($w_s = 1.0546$, $s = 0.9316$), $s_{max} = 0.93214$, which is greater than s, so the method is stable. For the pair ($w_s = 1.0743, s = 0.8001$), $s_{max} = 0.90986$, this method is also stable. At the time step size limit, the dispersion error is not the least with the values of the weight parameter found by search because the search criterion is for the anisotropy. This method is termed as the "high-accuracy-method by search" (HAM-S).

Using $w_s = 1.0546$ and $s = 0.9316$, the numerical dispersion from theory and numerical experiment is graphed in Fig.3-3 for HAM-S. The absolute numerical dispersion error is 0.247% along the axis, and 0.246% along the diagonal. The case of $w_s = 1.0743$ and $s = 0.8001$ was also numerically verified. The maximum dispersion error is 0.184%. Since the two cases have very close dispersion, the result is not shown.

The search method is a general way to find an "optimal value" of the weight parameter if the criterion of optimization is determined. The HAM method, particularly the HAM-S method, is a broadband, more efficient method, with a time-step size limit larger than the (2, 4) method. The IOM method is relatively broad band, and can have no numerical error at one designated frequency, as discussed in the next Section.

3.3 Coefficient-Modification Techniques

The IOM method can eliminate numerical anisotropy, but the residual dispersion error is large compared to the Yee's method. In order to reduce the numerical dispersion, many researchers modify the coefficients of a_1 and a_2 in the update equations. This thesis

refers such methods as the "coefficient-modification" techniques. In general, a_1 can be replaced by $a_1 a_{xe}$ and $a_1 a_{ye}$, and a_2 can be replaced by $a_2 a_{xh}$ and $a_2 a_{yh}$, respectively, where the parameters a_{xh}, a_{yh}, a_{xe} and a_{ye} can be modified differently. For example, References [36-37] use $a_{xe} = 1/\varepsilon_y$, $a_{ye} = 1/\varepsilon_x$ and $a_{xh} = a_{yh} = 1$; Reference [38] uses $a_{xh} = A$, $a_{yh} = B$ and $a_{xe} = a_{ye} = 1$ similar to Reference [98]. References [40-41] use $a_{xh} = a_{xe} = (\sin(\omega\Delta t/2)/\omega\Delta t)/(\sin(\beta_0\Delta x/2)/\beta_0\Delta x)$, $a_{yh} = a_{ye} = (\sin(\omega\Delta t/2)/\omega\Delta t)/(\sin(\beta_0\Delta y/2)/\beta_0\Delta y)$ in conjunction with adjacent cells. Reference [44] uses modification by solving equations from approximation of the dispersion relation. References [46-48] provide complicated filtering methods to modify these parameters. The parameter values are obtained according to different criteria, such as low numerical dispersion. All these methods change the "speed": in the standard FDTD, the physical speed is $c = 1/\sqrt{\mu\varepsilon}$; in the coefficient-modification techniques, the new speed is $c' = c\sqrt{a_{xh} a_{eh}}$ for a square mesh ($\Delta x = \Delta y$, $a_{xe} = a_{ye}$, $a_{xh} = a_{yh}$, $\varepsilon_x = \varepsilon_y$, $A = B$).

The different ways to put the a parameters into the update equations are equivalent in terms of numerical dispersion, but have different root-mean-square (RMS) errors [160] (Chapter 9). If the problem contains only one material, these techniques have the same effect. However, if the problem uses different materials, then unless all material parameters μ and ε are modified by the same factor, the reflection coefficient at a material boundary will change and there is artificial reflection. Therefore in coefficient modification the same factor for both a_1 and a_2 is preferable, that is, let $a_{xe} = a_{xh}$, and

$a_{ye} = a_{yh}$. For a square mesh, $a_{cm} = a_{xe} = a_{ye} = a_{xh} = a_{yh}$, they have the same magnitude, thus there is no artificial reflection.

A simple, straightforward way to find the values of the "a" parameters is to reduce or eliminate the numerical dispersion at one mesh density along one direction of travel. For example, to eliminate the axial numerical dispersion, the parameters are chosen to be

$$a_{cm} = \frac{\sin(\omega\Delta t/2)/\omega\Delta t}{w_{24i}\sin(\beta_m\Delta x/2)/\beta_m\Delta x + (1-w_{24i})\sin(3\beta_m\Delta x/2)/(3\beta_m\Delta x)} \tag{3-11}$$

Note that β_m is the theoretical phase constant at a designated frequency to have zero numerical dispersion, and is not necessarily the same as β_0 where zero anisotropy is desired. When they are the same, the combined IOM and the simple coefficient-modification technique can have zero anisotropy and zero numerical dispersion at one designated frequency. This method has been tested to reduce the numerical dispersion of IOM, and the results agree with theory very well.

The value of the "a" parameters can be assigned in different ways. However, since it has only one value for a given mesh density and time step size, it is possible to find an "optimal" value of the "a" parameters by searching once the sense of the optimization is set. Eqn. (3-11) is a special case to lead to zero numerical dispersion along axes.

3.4 Neighborhood-Averaging Method

The fact that a larger computational stencil can reduce numerical dispersion implies that nodes other than those used in the standard Yee stencil contribute "negative" dispersion. With proper weighting, better performance is obtained. The initial idea of the neighborhood-averaging method (NA) arises from the box method [69], and further investigation shows that it incorporates the collocated staggered formulation [128]. In this

thesis the "collocated" contribution is an average from nodes in the neighborhood of the nodes in the Yee's method. The weighted update equations in the 2D case are

$$\begin{aligned} H_z^{n+1/2}(i+1/2,j+1/2) &= H_z^{n-1/2}(i+1/2,j+1/2)+w_{na}a_2\{[E_x^n(i+1/2,j+1)-E_x^n(i+1/2,j)]/\Delta y \\ &-[E_y^n(i+1,j+1/2)-E_y^n(i,j+1/2)]/\Delta x\}+(1-w_{na})a_2\{[E_x^n(i+3/2,j+1) \\ &+E_x^n(i-1/2,j+1)-E_x^n(i+3/2,j)-E_x^n(i-1/2,j)]/2\Delta y-[E_y^n(i+1,j+3/2) \\ &+E_y^n(i,j-1/2)-E_y^n(i,j+3/2)-E_y^n(i,j-1/2)]/2\Delta x\} \end{aligned} \tag{3-12a}$$

$$\begin{aligned} E_x^{n+1}(i+1/2,j) &= E_x^n(i+1/2,j)+w_{na}a_1\{H_z^{n+1/2}(i+1/2,j+1/2)-H_z^{n+1/2}(i+1/2,j-1/2)\}/\Delta y \\ &+(1-w_{na})a_1\{H_z^{n+1/2}(i+3/2,j+1/2)+H_z^{n+1/2}(i-1/2,j+1/2)- \\ &H_z^{n+1/2}(i+3/2,j-1/2)-H_z^{n+1/2}(i-1/2,j-1/2)\}/2\Delta y \end{aligned} \tag{3-12b}$$

$$\begin{aligned} E_y^{n+1}(i,j+1/2) &= E_y^n(i,j+1/2)-w_{na}a_1\{H_z^{n+1/2}(i+1/2,j+1/2)-H_z^{n+1/2}(i-1/2,j+1/2)\}/\Delta x \\ &+(1-w_{na})a_1\{[H_z^{n+1/2}(i+1/2,j+3/2)+H_z^{n+1/2}(i+1/2,j-1/2)- \\ &H_z^{n+1/2}(i-1/2,j+3/2)-H_z^{n+1/2}(i-1/2,j-1/2)\}/2\Delta x \end{aligned} \tag{3-12c}$$

The optimal value of w_{na} is to be determined to optimize the dispersion behavior in some sense. Setting $w_{na}=1$ recovers the usual Yee update equations. Setting $w_{na}=0$ obtains the collocated-staggered formulation [128]. The Yee's method has smallest numerical velocity along the axes, and the collocated method has the highest numerical velocity along the axes [128]. Thus their proper combination should give better performance.

The amplification factor can be obtained from (3-12a), (3-12b) and (3-12c) as

$$\begin{aligned} \xi &= 1-2(r_x^2(w_{na}+(1-w_{na})p_{xna})^2+r_y^2(w_{na}+(1-w_{na})p_{yna})^2) \\ &\pm J\sqrt{1-(1-2(r_x^2(w_{na}+(1-w_{na})p_{xna})^2+r_y^2(w_{na}+(1-w_{na})p_{yna})^2))^2} \\ &= \exp\left(\pm J\tan^{-1}\frac{\sqrt{1-(1-2(r_x^2(w_{na}+(1-w_{na})p_{xna})^2+r_y^2(w_{na}+(1-w_{na})p_{yna})^2))^2}}{1-2(r_x^2(w_{na}+(1-w_{na})p_{xna})^2+r_y^2(w_{na}+(1-w_{na})p_{yna})^2)}\right) \end{aligned} \tag{3-13}$$

where $p_{xna}=\cos(\beta_y\Delta y)$, $p_{yna}=\cos(\beta_x\Delta x)$. The dispersion relation can be derived from the amplification factor as

$$\left(\frac{\sin(\omega\Delta t/2)}{c\Delta t}\right)^2 = \frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}[w_{na} + (1-w_{na})\cos(\beta_y\Delta y)]^2 + \frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}[w_{na} + (1-w_{na})\cos(\beta_x\Delta x)]^2 \tag{3-14}$$

Similar to the O24 method, the value of the weight parameter for the neighborhood-averaging-diagonal" (NAD) method can be obtained as

$$w_{nad} = \frac{\left(\frac{\sin(\omega\Delta t/2)}{\sqrt{2}c\Delta t/\Delta x}/\frac{\sin(\beta_0\Delta x\sqrt{2}/4)}{\Delta x} - \cos(\beta_0\Delta x\sqrt{2}/2)\right)}{\left(1-\cos(\beta_0\Delta x\sqrt{2}/2)\right)} \tag{3-15}$$

The numerical dispersion from the theory (solid line) and the numerical experiments (small circles) optimized at 10 CPW is shown in Fig. 3-4 with good agreement.

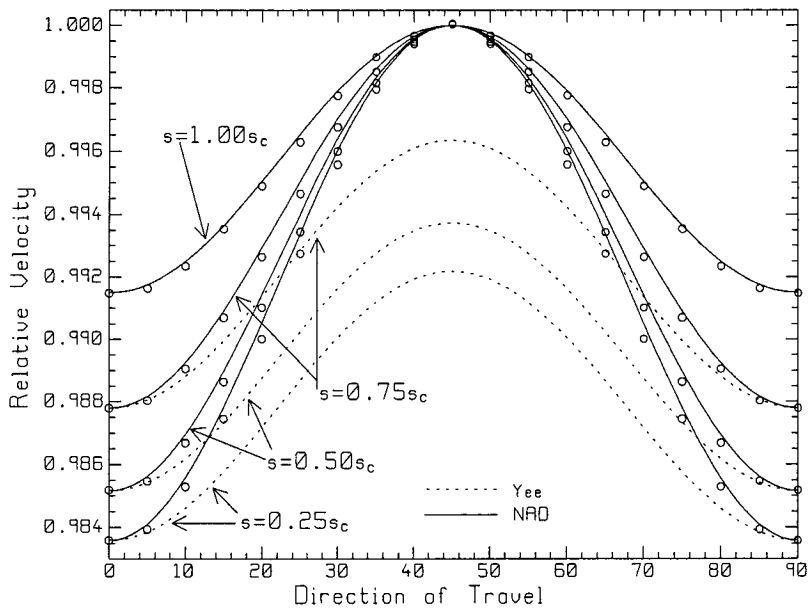

Fig. 3-4 Numerical dispersion of the NAD method optimized at 10 CPW.

The NA method can also eliminate the numerical anisotropy, termed as the "neighborhood-averaging-isotropic" (NAI) method. To have zero anisotropy, the weight parameter w_{na} can be derived as

$$w_{nai} = \frac{\left(\frac{\sin(\beta_0 \Delta x / 2)}{\sqrt{2}\sin(\beta_0 \Delta x \sqrt{2} / 4)} - \cos(\beta_0 \Delta x \sqrt{2} / 2) \right)}{\left(1 - \cos(\beta_0 \Delta x \sqrt{2} / 2)\right)} \tag{3-16}$$

The value of w_{nai} is independent of the time step size, and increases slowly as the optimized mesh density increases, and approaches 11/12 with a very fine mesh. Numerical experiments have been done with optimization at 10, 20 and 30 CPW. Fig. 3-5 shows the theoretical predication using (3-16) with solid lines and numerical results as small circles for various Courant numbers, optimized at mesh density 20 CPW. The theory and the numerical experiments agree with each other quite well. This method has a larger time-step-size limit than Yee's FDTD, and has a smaller numerical velocity than the speed of light. In contrast, the IOM method in Fig. 3-2 produces a numerical velocity larger than the speed of light. The next section will present a new method combining the IOM method with the NAI method, leading to much higher numerical accuracy.

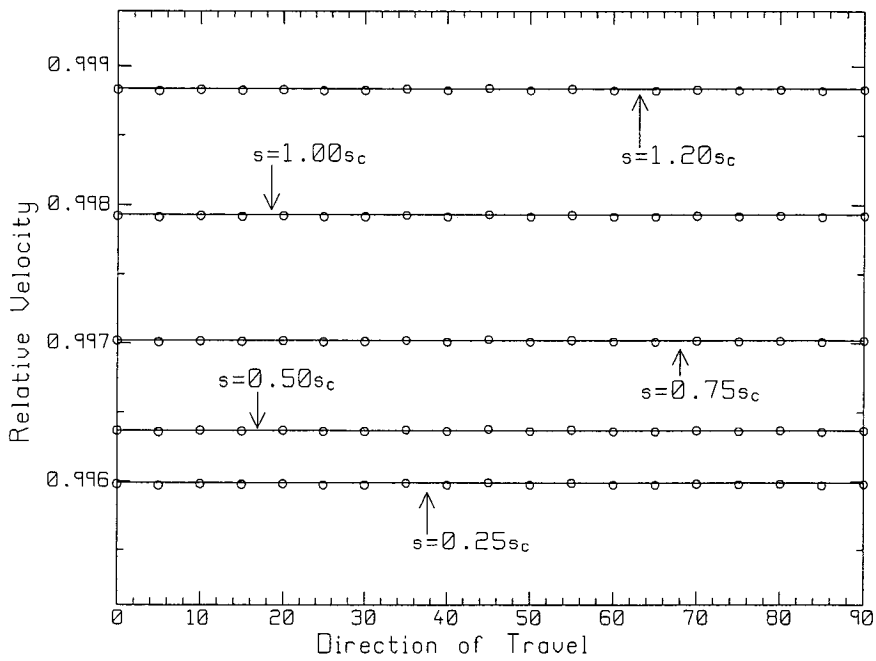

Fig. 3-5 Numerical dispersion of the NAI method optimized at mesh density 20 CPW and the Yee's method.

3.5 Optimized Neighborhood-Averaging 24 Stencil Method

The IOM and the NAI methods proposed in the preceding sections form the basis of the "neighborhood-averaging 24" stencil (NA24) method. For conciseness, first construct some central finite-difference (FD) operators for the first-degree spatial derivatives. The central FD operator D_c, the one-cell-away FD operator D_{24} based on the (2,4) stencil, and the neighborhood-averaging FD operator D_{na} discussed above are defined for derivatives along the x-axis as

$$D_{xc} = \frac{f(x+\Delta x/2, y) - f(x-\Delta x/2, y)}{\Delta x} \tag{3-17a}$$

$$D_{x24} = \frac{f(x+3\Delta x/2, y) - f(x-3\Delta x/2, y)}{3\Delta x} \tag{3-17b}$$

$$D_{xna} = \frac{f(x+\Delta x/2, y+\Delta y) + f(x+\Delta x/2, y-\Delta y) - f(x-\Delta x/2, y+\Delta y) - f(x-\Delta x/2, y-\Delta y)}{2\Delta x} \tag{3-17c}$$

where f is the function. Similarly, the FD operators for y-derivatives are defined as

$$D_{yc} = \frac{f(x, y+\Delta y/2) - f(x, y-\Delta y/2)}{\Delta y} \tag{3-18a}$$

$$D_{y24} = \frac{f(x, y+3\Delta y/2) - f(x, y-3\Delta y/2)}{3\Delta y} \tag{3-18b}$$

$$D_{yna} = \frac{f(x+\Delta x, y+\Delta y/2) + f(x+\Delta x, y-\Delta y/2) - f(x-\Delta x, y+\Delta y/2) - f(x-\Delta x, y-\Delta y/2)}{2\Delta y} \tag{3-18c}$$

The center of the difference is at (x, y).The new weighted FD operators for the first-degree derivatives along x-axis and along y-axis can be built up as

$$D_x = w\left(w_{24i} D_{xc} + (1-w_{24i}) D_{x24}\right) + (1-w)\left(w_{nai} D_{xc} + (1-w_{nai}) D_{xna}\right) \tag{3-19a}$$

$$D_y = w\left(w_{24i} D_{yc} + (1-w_{24i}) D_{y24}\right) + (1-w)\left(w_{nai} D_{yc} + (1-w_{nai}) D_{yna}\right) \tag{3-19b}$$

where w is the weight parameter to be optimized. The new FD operators (3-19) are a linear combination of other FD operators with second-order accuracy. Thus they have the second-order accuracy in space, and are simpler in conception than those in [44-48].

Using these constructed FD operators, the NA24 FDTD method is formulated as

$$H_z^{n+1/2} = H_z^{n-1/2} + a_2\left(D_y E_x^n - D_x E_y^n\right) \tag{3-20a}$$

$$E_x^{n+1} = E_x^n + a_1 D_y H_z^{n+1/2} \tag{3-20b}$$

$$E_y^{n+1} = E_y^n - a_1 D_x H_z^{n+1/2} \tag{3-20c}$$

The amplification factor can be obtained as

$$\xi = e^{\pm J \tan^{-1}\frac{2\sqrt{(Q_x^2+Q_y^2)(1-Q_x^2-Q_y^2)}}{1-(Q_x^2+Q_y^2)}} \tag{3-21}$$

where

$$Q_x = r_x\left(w\left(w_{24i} + (1-w_{24i})p_{x24}\right) + (1-w)\left(w_{nai} + (1-w_{nai})p_{xna}\right)\right) \tag{3-22a}$$

$$Q_y = r_y\left(w\left(w_{24i} + (1-w_{24i})p_{y24}\right) + (1-w)\left(w_{nai} + (1-w_{nai})p_{yna}\right)\right) \tag{3-22b}$$

The numerical dispersion relation can be derived as

$$\left(\frac{\sin(\omega\Delta t/2)}{c\Delta t}\right)^2 = \frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\left[w(w_{24i} + (1-w_{24i})p_{x24}) + (1-w)(w_{nai} + (1-w_{nai})p_{xna})\right]^2 + \frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\left[w(w_{24i} + (1-w_{24i})p_{y24}) + (1-w)(w_{nai} + (1-w_{nai})p_{yna})\right]^2 \tag{3-23}$$

Similar to the isotropic methods discussed above, one can derive a formula to find the optimal value of the weight parameter w. However, since both w_{24i} and w_{nai} are highly isotropic, the formula suffers from the subtraction cancellation of precision. Numerical experiments show that use of such a formula does not lead to a satisfactory result. An alternate approach to determine the optimal value of w is to use a search method according to Eqn. (3-23). The idea is that, for a given optimized mesh density, increase the parameter w at a step of one thousandth, and calculate the numerical dispersion error along the axis. The search stops when the numerical error changes sign from negative to positive. The weight parameter can be found by a linear interpolation. Note that near the optimal value of the weight parameter, the difference between the dispersion errors along

the axis and the diagonal is less than 10^{-6}. One can use this difference as a second criterion in the search method to ensure the isotropy. In addition, one can use the optimal value of w found using the search method to calculate the numerical dispersion error and anisotropy to verify the correctness. For example, when optimized at 10 CPW, the search method gives $w = 0.487447$ at the time step size of the Courant limit. Fig. 3-6 shows the numerical experiments for the NA24 method optimized at 10 CPW and also optimized at each of the four time step sizes: $s = s_c$ (circle), $s = 0.75 s_c$ (triangle), $s = 0.5 s_c$ (square) and $s = 0.25 s_c$ (cross), The figure demonstrates that the NA24 method has zero anisotropy and zero numerical dispersion error within the accuracy limit of the matching method. For comparison, the numerical dispersion of Yee's FDTD using the Courant time step size limit is also shown in Fig. 3-6.

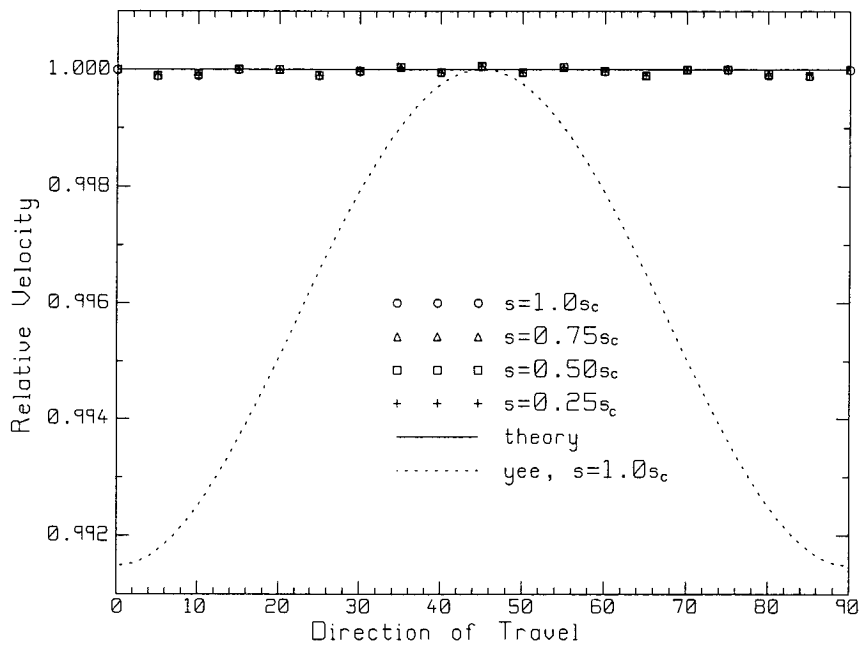

Fig. 3-6 Numerical dispersion of the NA24 method and the Yee's FDTD.

Although the optimal values of the weight parameters for IOM and NAI do not depend on the time step size, their residual dispersion error does depend on the time step

size. Therefore the optimal value of the weight parameter w will depend on the time step size, because it makes the dispersion disappear for a given time step size.

3.6 Discussion

3.6.1 Time step size limit

The three methods proposed above are explicit, thus like Yee's method, they also have an upper-bound of the time step size limit. From their amplification factors, the time step size limits can be derived as follows [159]

$$\Delta t_{24} \le \frac{3}{4w_{24}-1}\Delta t_c \tag{3-24a}$$

$$\Delta t_{na} \le \frac{1}{2w_{na}-1}\Delta t_c \tag{3-24b}$$

$$\Delta t \le \frac{1}{w(4w_{24}-1)/3+(1-w)(2w_{na}-1)}\Delta t_c \tag{3-24c}$$

When optimized at 10 CPW, the time step size limit Δt_{24} is 0.84867 times Δt_c for IOM with $w_{24} = w_{24i} = 1.133733$; for HAM-S method, if $w_{24} = w_s = 1.0546$, the limit is 0.93214; for NAI with $w_{na} = w_{nai} = 0.915212$, Δt_{na} is 1.20420 times Δt_c, respectively. For HAM-FW method, $w_{24} = 27/26$, the limit is $39/41$, which is larger than that of the standard (2,4) method. For the NA24 method, the search algorithm finds that $w = 0.487447$, and the time step size limit is the same as the Courant limit using (3-24c). Numerical experiments have verified the validity of (3-24).

It is interesting to note that the time step size limit of the NA24 method is a weighted "parallel" combination of the limits for O24 and NA method, that is,

$$\frac{1}{\Delta t_{na24}} = w\frac{1}{\Delta t_{24}} + (1-w)\frac{1}{\Delta t_{na}} \tag{3-25}$$

Generally Δt_{24} is smaller than Δt_c, and Δt_{na} is larger than Δt_c. However, since w is a function of time step size, when the search method is used to find a proper value of w, one must verify using Eqn. (3-25) that the time step size used is within the limit.

For AOM and DOM, since they have the optimal value of the weight parameter smaller than that of IOM [159], their time step size limit for large time step size (e.g., $s > 0.8 s_c$) is larger than that of IOM and (2,4) method. Particularly, the limit for DOM is close to the Courant limit.

3.6.2 The order of accuracy

By Taylor Series analysis, the optimized methods in this Chapter are 2nd order accurate both in time and space, and the standard (2,4) method is 2nd order accurate in time and 4th order accurate in space. However, analysis shows [159] that the (2, 4) method is only 2nd order accurate in terms of numerical dispersion, and 4th order accurate in terms of anisotropy. Because of the partial cancellation of the higher-order terms, the numerical error of the proposed optimized methods is compensated to a certain extent. Thus they have higher than 2nd order accuracy. This can be explained because the spatial accuracy is also affected by the Courant number used [159]. This fact implies that if the accuracy orders in time and space are not compatible, the potentially-high accuracy cannot be fully utilized. This is the part of the reason that this thesis proposes 2nd order-accurate optimized methods.

3.6.3 Accumulated phase error of the NA24 method

This section only evaluates the most accurate NA24 method. The maximum accumulated-phase error per cell of wave travel versus the mesh density is shown in Fig. 3-7. The maximum accumulated-phase error occurs at 13 CPW and is 0.002367° per

cell. For comparison, the maximum accumulated-phase error is larger than 0.05° per cell for the methods in [44] [47-48]. The proposed NA24 method has more than 20 times higher accuracy. To maintain the accumulated-phase error less than 0.002367° per cell, the minimum mesh density can be as low as 9.4 CPW. Thus the highest frequency is $c/(9.4\Delta x)$. Any frequency lower than $c/(9.4\Delta x)$ has smaller accumulated phase error. Therefore the NA24 method provides better accuracy over a broad bandwidth.

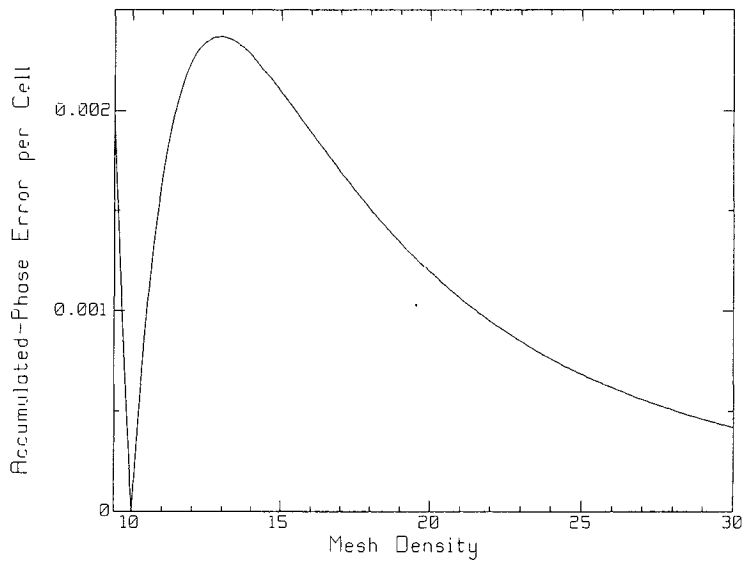

Fig. 3-7 The Maximum accumulated-phase error per cell distance for NA24 method.

3.6.4 3D NA24 method

It is straightforward to extend the proposed methods in this Chapter to the 3D case. The O24 FD operator does not change, but needs one more difference along the z-coordinate axis. The NA FD operator requires a modification to make a full average of the neighborhood. The 3D NA FD operators centered at (x, y, z) are

$$\begin{aligned} D_{xna}=[&f(x+\Delta x/2,y+\Delta y,z+\Delta z)+f(x+\Delta x/2,y-\Delta y,z+\Delta z)+f(x+\Delta x/2,y+\Delta y,z-\Delta z)\\ &+f(x+\Delta x/2,y-\Delta y,z-\Delta z)-f(x-\Delta x/2,y+\Delta y,z+\Delta z)-f(x-\Delta x/2,y-\Delta y,z+\Delta z)\\ &-f(x-\Delta x/2,y+\Delta y,z-\Delta z)-f(x-\Delta x/2,y-\Delta y,z-\Delta z)]/4\Delta x \end{aligned} \tag{3-26a}$$

$$D_{yna}=[f(x+\Delta x,y+\Delta y/2,z+\Delta z)+f(x+\Delta x,y+\Delta y/2,z-\Delta z)+f(x-\Delta x,y+\Delta y/2,z+\Delta z) +f(x-\Delta x,y+\Delta y/2,z-\Delta z)-f(x+\Delta x,y-\Delta y/2,z+\Delta z)-f(x+\Delta x,y-\Delta y/2,z-\Delta z) -f(x-\Delta x,y-\Delta y/2,z+\Delta z)-f(x-\Delta x,y-\Delta y/2,z-\Delta z)]/4\Delta y \tag{3-26b}$$

$$D_{zna}=[f(x+\Delta x,y+\Delta y,z+\Delta z/2)+f(x+\Delta x,y-\Delta y,z+\Delta z/2)+f(x-\Delta x,y+\Delta y,z+\Delta z/2) +f(x-\Delta x,y-\Delta y,z+\Delta z/2)-f(x+\Delta x,y+\Delta y,z-\Delta z/2)-f(x+\Delta x,y-\Delta y,z-\Delta z/2) -f(x-\Delta x,y+\Delta y,z-\Delta z/2)-f(x-\Delta x,y-\Delta y,z-\Delta z/2)]/4\Delta z \tag{3-26c}$$

The amplification factor for 3D NA24 method is similar to Eqn. (3-12), and given by

$$\xi = \exp\left(\pm J \tan^{-1} \frac{2\sqrt{(Q_x^2+Q_y^2+Q_z^2)(1-Q_x^2-Q_y^2-Q_z^2)}}{1-(Q_x^2+Q_y^2+Q_z^2)} \right) \tag{3-27}$$

where

$$Q_x = r_x\left(w\left(w_{24i}+(1-w_{24i})p_{x24}\right)+(1-w)\left(w_{nai}+(1-w_{nai})p_{xna}\right)\right) \tag{3-28a}$$

$$Q_y = r_y\left(w\left(w_{24i}+(1-w_{24i})p_{y24}\right)+(1-w)\left(w_{nai}+(1-w_{nai})p_{yna}\right)\right) \tag{3-28b}$$

$$Q_z = r_z\left(w\left(w_{24i}+(1-w_{24i})p_{z24}\right)+(1-w)\left(w_{nai}+(1-w_{nai})p_{zna}\right)\right) \tag{3-28c}$$

$$p_{xna} = \cos(\beta_y \Delta y)\cos(\beta_z \Delta z) \tag{3-29a}$$

$$p_{yna} = \cos(\beta_x \Delta x)\cos(\beta_z \Delta z) \tag{3-29b}$$

$$p_{zna} = \cos(\beta_x \Delta x)\cos(\beta_y \Delta y) \tag{3-29c}$$

The numerical dispersion relation is similar to Eqn. (3-23), and given as

$$\left(\frac{\sin(\omega\Delta t/2)}{c\Delta t}\right)^2 = Q_x^2+Q_y^2+Q_z^2 \tag{3-30}$$

The expression for the time step size is the same as Eqn. (3-24c).

3.6.5 Non-square mesh

The NA24 method can also be used in a non-square mesh with some modification. Because the spatial increments along the axes are not the same for the non-square mesh, the numerical dispersion errors along different axes are not the same, and the direction of travel with the highest numerical velocity changes with respect to the aspect ratio. To apply the NA24 method, first the coefficient-modification technique is used to

compensate the numerical dispersion error along other axes in order to have the same dispersion as that of the finest spatial increment. Then the optimal values of the weight parameters for the optimized (2,4) stencil method, the NA method and the NA24 method can be found accordingly. The time step size limit may be a little smaller than that for a square mesh due to the coefficient correction.

3.7 Summary

For problems in which the wave is confined to travel within a known sector space, the AOM, DOM, or NAD methods can be used with high accuracy, without complicated filtering. The high-accuracy methods with fixed weight parameter (HAM-FW) have a constant value of the weight parameter, similar to (2,4) method. The search method (HAM-S) is a general method for finding the optimal value of the weight parameter depending on the sense of optimization. Those methods can be incorporated into the current (2,4) method without extra computational resources. The IOM and NAI methods can eliminate numerical anisotropy, and numerical dispersion if the coefficient-modification technique is used, without consuming additional computational resources. The NA24 method provides much higher (more than 20 times) accuracy than some published methods with about the same time step size limit of Yee's FDTD. It is a broadband, high accuracy method.

For problems with very fine geometrical features to be resolved, explicit methods are not efficient due to the upper-bound time step size limit, which is common for all explicit methods in solving Maxwell's Equations. In the next three Chapters, several new methods without the time step size limits are proposed to alleviate this limitation.

Chapter 4 Numerical Dissipation and Dispersion in ADI-FDTD

With technology advances, the clock frequency and the density of integrated circuits go higher and higher. In simulating such devices, explicit FDTD methods suffer from severe inefficiency. For a high-speed VLSI interconnect with a minimum geometrical feature size of 1 μm and a signal rise time less than 100ps [85], the frequency spectrum of this signal extends to nearly 10 GHz. The resulting minimum wavelength is on the order of 3 cm in free space, or 1.5 cm in SiO_2 (ϵ_r=4). In order to keep proper numerical accuracy, the appropriate grid size is on the order of 1.0 mm (between 10 to 20 cells per wavelength). However, in order to model the micron-scale features of the interconnect, the spatial increment must be on the order of 1.0 μm or less. As a result, the upper bound on the time increment is limited approximately only 2 fs. Simulating the 100-ps rise time of digital signal alone would need 50,000 time steps! To alleviate the problem, the unconditionally-stable FDTD methods which can use a larger time step size are desired.

4.1 Introduction

Among the unconditionally-stable FDTD methods, the alternating-direction implicit (ADI) method for FDTD (ADI-FDTD) [81-83] [85] is the algorithm most investigated in recent years. However, there is inconsistency [152] in the literature concerning the numerical dispersion relation [81], [99-100]. Later several papers address this problem [101-103] [152-154]. Most papers [99] [101-103] derive the dispersion relation using the matrix method, which is correct. However, since the overall ADI-FDTD scheme is dissipation-free, the growth and dissipation mechanism of the individual sub-marching

procedures of ADI-FDTD cannot be revealed by the matrix method. This thesis uses the Fourier method with the amplification factor for each ADI sub-marching procedure, and shows that the exact cancellation of the growth and dissipation makes the overall ADI-FDTD unconditionally stable [152][154]. To account for the dissipation, the imaginary part of the frequency in the plane wave expression must be used.

All the unconditionally-stable methods proposed in this thesis are based on the Crank-Nicolson (CN) scheme, thus first the application of the CN scheme to solve Maxwell's Equations will be introduced and different forms of the CN scheme will be given for later comparison. After formulating ADI-FDTD from the CN scheme, the numerical dispersion relations for 1D and 2D ADI-FDTD will be re-derived and validated. The perturbation terms of ADI to CN will be given. A higher accuracy ADI method is also proposed by use of the (2, 4) stencil.

4. 2 Various Forms of Crank-Nicolson Scheme

4.2.1 Crank-Nicolson scheme and the numerical dispersion relation

The Crank-Nicloson (John Crank, 1916- , Phyllis Nicolson, 1917-1968) scheme is a well-known implicit numerical method [49] [69] [129]. It balances the right-hand-side (RHS) of the difference equations using an average at the time step n and time step n+1 to maintain 2nd order accuracy in time. For a 2D TE_z wave in a linear, isotropic, non-dispersive and lossless medium, the CN scheme can be written [151]

$$E_x^{n+1}(i+1/2,j)=E_x^n(i+1/2,j) + a_1\frac{H_z^{n+1}(i+1/2,j+1/2)-H_z^{n+1}(i+1/2,j-1/2)+H_z^n(i+1/2,j+1/2)-H_z^n(i+1/2,j-1/2)}{\Delta y} \quad (4\text{-}1a)$$

$$E_y^{n+1}(i,j+1/2)=E_y^n(i,j+1/2) - a_1\frac{H_z^{n+1}(i+1/2,j+1/2)-H_z^{n+1}(i-1/2,j+1/2)+H_z^n(i+1/2,j+1/2)-H_z^n(i-1/2,j+1/2)}{\Delta x} \quad (4\text{-}1b)$$

$$H_z^{n+1}(i+1/2,j+1/2) = H_z^n(i+1/2,j+1/2)$$
$$+a_2\frac{E_x^{n+1}(i+1/2,j+1)-E_x^{n+1}(i+1/2,j)+E_x^n(i+1/2,j+1)-E_x^n(i+1/2,j)}{\Delta y}$$
$$-a_2\frac{E_y^{n+1}(i+1,j+1/2)-E_y^{n+1}(i,j+1/2)+E_y^n(i+1,j+1/2)-E_y^n(i,j+1/2)}{\Delta x} \quad (4\text{-}1c)$$

where $a_1 = \Delta t/2\varepsilon$, $a_2 = \Delta t/2\mu$. It can be seen that the CN scheme is 2nd order accurate in time and space. Using the Fourier analysis, the amplification coefficient matrix for the CN scheme can be written as

$$Q=\begin{pmatrix} 1-\xi & 0 & -J2a_1(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y \\ 0 & 1-\xi & J2a_1(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x \\ -J2a_2(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y & J2a_2(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x & 1-\xi \end{pmatrix} \quad (4\text{-}2)$$

From the determinant of Q, the amplification polynomial can be obtained as

$$P(\xi) = (1+r_x^2+r_y^2)(\xi^3-1)+(3+r_x^2+r_y^2)(\xi-\xi^2) \quad (4\text{-}3)$$

Eliminating the stationary mode solution, the amplification factor can be found from the roots of the amplification polynomial as

$$\xi = e^{\pm J\tan^{-1}\frac{\sqrt{(1+r_x^2+r_y^2)^2-(1-r_x^2-r_y^2)^2}}{1-(r_x^2+r_y^2)}} \quad (4\text{-}4)$$

Since the amplification factor has a unity magnitude, the CN scheme is unconditionally stable. The numerical dispersion relation can be derived [151] as

$$\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2} = \frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} \quad (4\text{-}5)$$

Since the tangent is always larger than the sine of Yee's FDTD Eqn. (2-12), for the same parameters, the CN scheme has larger numerical error than Yee's method even when the Courant number is smaller than the Courant limit. In addition, the CN scheme requires a huge, sparse, banded matrix be solved at each time step, which demands large

computer resources. Thus, to efficiently implement the CN scheme, some modifications must be made. The simplest form which maintains the unconditional stability requires to only solve tridiagonal matrices.

4.2.2 Other forms of the CN scheme

This thesis will develop other unconditionally-stable methods by modifying the CN scheme, and the proposed new methods will be compared to CN. For convenience, the CN scheme will be written in various different forms. For brevity, the central-difference operators D_x and D_y for the first derivatives, as well as the difference operators D_{2x} and D_{2y} for the second derivatives are used. Inserting Eqs. (4-1a) and (4-1b) into Eqn. (4-1c) obtains [151]

$$\left\{1-b^2(D_{2x}+D_{2y})\right\}H_z^{n+1}=\left\{1+b^2(D_{2x}+D_{2y})\right\}H_z^n+2a_2(D_yE_x^n-D_xE_y^n) \tag{4-6}$$

where $b=c\Delta t/2$. Inserting Eqn. (4-1c) into Eqs. (4-1a) and (4-1b) obtains [157]

$$(1-b^2D_{2y})E_x^{n+1}+b^2D_xD_yE_y^{n+1}=(1+b^2D_{2y})E_x^n-b^2D_xD_yE_y^n+2a_1D_yH_z^n \tag{4-7a}$$

$$(1-b^2D_{2x})E_y^{n+1}+b^2D_xD_yE_x^{n+1}=(1+b^2D_{2x})E_y^n-b^2D_xD_yE_x^n-2a_1D_xH_z^n \tag{4-7b}$$

Eqs. (4-7a) and (4-7b) can be further solved to be

$$\left\{1-b^2(D_{2x}+D_{2y})\right\}E_x^{n+1}=\left\{1-b^2(D_{2x}-D_{2y})\right\}E_x^n-2b^2D_xD_yE_y^n+2a_1D_yH_z^n \tag{4-8a}$$

$$\left\{1-b^2(D_{2x}+D_{2y})\right\}E_y^{n+1}=\left\{1+b^2(D_{2x}-D_{2y})\right\}E_y^n-2b^2D_xD_yE_x^n-2a_1D_xH_z^n \tag{4-8b}$$

These forms Eqs. (4-6), (4-7) and (4-8) are derived from the original CN formula Eqn. (4-1) without adding or dropping any extra terms, thus they have the same accuracy and stability as Eqn. (4-1).

4.2.3 Formulation of ADI from the CN scheme

The CN scheme (1) can be written in a matrix form as

$$(I-D_1-D_2)W^{n+1}=(I+D_1+D_2)W^n \tag{4-9}$$

where I is 3x3 identity matrix, $W=\begin{pmatrix}E_x & E_y & H_z\end{pmatrix}^T$, D_1 and D_2 are spatial difference matrices given by

$$D_1=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & -a_1D_x\\ 0 & -a_2D_x & 0\end{bmatrix} \qquad D_2=\begin{bmatrix}0 & 0 & a_1D_y\\ 0 & 0 & 0\\ a_2D_y & 0 & 0\end{bmatrix} \tag{4-10}$$

By adding $D_1D_2W^{n+1}$ to the left-hand-side (LHS) and $D_1D_2W^n$ to the right-hand-side (RHS) of Eqn. (4-9), Eqn. (4-9) can be approximately factorized as

$$(I-D_1)(I-D_2)W^{n+1}=(I+D_1)(I+D_2)W^n \tag{4-11}$$

This method is called "approximation factorization" [77]. It can be seen that adding the term $D_1D_2W^{n+1}-D_1D_2W^n$ is for factorization, thus the term $D_1D_2W^{n+1}-D_1D_2W^n$ can be called "factorization error". It can be shown that the factorization error added to Eqn. (4-1b) is $-b^2D_xD_y(E_x^{n+1}-E_x^n)$, there is no factorization error in the other two equations. If the positions of D_1 and D_2 are exchanged, the only factorization error added is $-b^2D_xD_y(E_y^{n+1}-E_y^n)$ to Eqn. (4-1a). Since the added terms are higher-order, they do not change the stability as mentioned in chapter 2. Therefore Eqn. (4-11) is unconditionally stable. Note that such formulation does not require that D_1 and D_2 commute.

To solve Eqn. (4-11), it can be split into two time steps at n+1/2 and n+1 as

$$(I-D_1)W^{n+1/2}=(I+D_2)W^n \tag{4-12a}$$

$$(I-D_2)W^{n+1}=(I+D_1)W^{n+1/2} \tag{4-12b}$$

where n+1/2 is intermediate time step. The implicit operation is alternated from the x-coordinate (D_1) to the y-coordinate (D_2), thus is called "alternating-direction implicit" (ADI) method [49] [76-78]. It is a two-step time-marching scheme different from Yee's leap-frog time marching scheme.

Note that in Eqn. (4-12) the field values at the intermediate time step n+1/2 are non-physical; thus they cannot be interpreted to be the actual field values. In other words, if ADI is hybridized with other one-step methods by embedding an ADI fine mesh to a coarse mesh, say, Yee's FDTD [88], ADI must be performed for a full update cycle from time step n to n+1. Otherwise, the non-physical field values will propagate erroneously into the enclosing mesh [154].

4.3 Numerical Dissipation and Dispersion of 1D ADI-FDTD

Following the formulation of Eqn. (4-12) and collapsing to 1D, the update equations for the two sub-marching procedures of one-dimensional ADI-FDTD are [154]

$$E_y^{n+1/2}(i) = E_y^n(i) - a_1 \frac{H_z^{n+1/2}(i+1/2) - H_z^{n+1/2}(i-1/2)}{\Delta x} \tag{4-13.1a}$$

$$H_z^{n+1/2}(i+1/2) = H_z^n(i+1/2) - a_2 \frac{E_y^{n+1/2}(i+1) - E_y^{n+1/2}(i)}{\Delta x} \tag{4-13.1b}$$

$$E_y^{n+1}(i) = E_y^{n+1/2}(i) - a_1 \frac{H_z^{n+1/2}(i+1/2) - H_z^{n+1/2}(i-1/2)}{\Delta x} \tag{4-13.2a}$$

$$H_z^{n+1}(i+1/2) = H_z^{n+1/2}(i+1/2) - a_2 \frac{E_y^{n+1/2}(i+1) - E_y^{n+1/2}(i)}{\Delta x} \tag{4-13.2b}$$

Note that $E_y^{n+1/2}$ and $H_z^{n+1/2}$ are non-physical intermediate values as mentioned before. After inserting the plane wave solution into the Eqs. (4-13.1) and (4-13.2), the two amplification polynomials for the two sub-marching procedures can be obtained as

$$P(\xi_1) = 1 - 2\xi_1 + (1 + r_x^2)\xi_1^2 \tag{4-14a}$$

$$P(\xi_2) = 1 + r_x^2 - 2\xi_2 + \xi_2^2 \tag{4-14b}$$

The two amplification factors for each sub-marching procedure can be obtained from the roots of Eqn. (4-14) as [154] [160]

$$\xi_1 = \xi_{10} e^{\pm J \tan^{-1}(s \sin(\beta \Delta x/2))} \qquad \xi_2 = \xi_{20} e^{\pm J \tan^{-1}(s \sin(\beta \Delta x/2))} \tag{4-15a}$$

$$\xi_{10} = \frac{1}{\sqrt{1 + s^2 \sin^2(\beta \Delta x/2)}} = \frac{1}{\xi_{20}} \tag{4-15b}$$

Eqn. (4-15.b) shows that the first sub-step is unconditionally stable because the magnitude of the amplification factor $|\xi_1| = \xi_{10}$ is always smaller than unity, and the second sub-step is not stable because the magnitude of the amplification factor $|\xi_2| = \xi_{20}$ is always larger than unity. Therefore neither (4-13.1) nor (4-13.2) can be used alone as a numerical scheme because of exponential dissipation or growth. Formulating Eqn. (4-13.1) as an unconditionally-stable FDTD has proved the dissipation [154]. By using the complex angular frequency [4] [49] the two amplification factors can be expressed in terms of the time step size as

$$\xi_1 = e^{-\omega_{im1}\Delta t/2} e^{j\omega \Delta t/2} \tag{4-16a}$$

$$\xi_2 = e^{-\omega_{im2}\Delta t/2} e^{j\omega \Delta t/2} \tag{4-16b}$$

where ω_{im1} is dissipation coefficient and ω_{im2} is growth coefficient, given by [154]

$$\omega_{im1} = -\omega_{im2} = \frac{\ln[1 + s^2 \sin^2(\beta \Delta x/2)]}{\Delta t} = \frac{\ln[1/\cos^2(\omega \Delta t/2)]}{\Delta t} \tag{4-16c}$$

The dissipation and the growth coefficients have the same magnitude but opposite sign, thus they cancel each other over one complete update cycle. This dissipation is an "artificial viscosity". The dissipation and growth come from the "unbalanced" time formulation in the RHSs of the Maxwell's Equations. In other words, the time

discretization at each sub-marching procedure is 1st order, thus it causes dissipation or growth, as described in Chapter 2.

The numerical dispersion equation for 1D ADI-FDTD is given in [154] [160] as

$$\tan^2(\omega\Delta t/2) = s^2 \sin^2(\beta\Delta x/2) \tag{4-17}$$

It can be seen from Eqn. (4-15a) that the phases of ξ_1 and ξ_2 are identical. Therefore the numerical dispersion relation is the same for both sub-steps and thus (4-17) is valid for the whole algorithm. The hypothesis that each individual sub-marching step has different numerical dispersion relation [100] is not correct. Note that (4-17) can also be obtained with the amplification matrix method in [160] and is equivalent to that in [99].

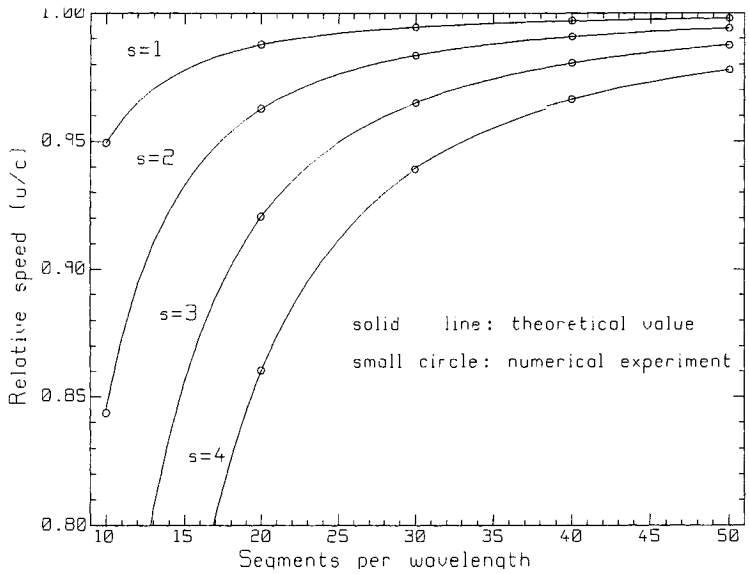

Fig.4-1 Numerical dispersion as a function of mesh density in 1D.

Fig. 4-1 graphs the numerical relative wave velocity as a function of the mesh density $N = \lambda/\Delta x$, where $\lambda = 1$ m, for s=1, 2, 3 and 4, according to Eqn. (4-17). The error decreases rapidly as N increases and increases dramatically as s becomes larger. At the Courant limit, s=1, the wave velocity is in error by about 0.2% for N=50 CPW. As the

Courant number increases, the error increases. For example, s=4, the wave velocity is in error by about 2.18% for N=50.

Numerical experiments were performed to verify that the dispersion predicted by (4-17) is correct. A cell space with 5000 cells was used. The "close" observer was at cell number 2600 and the "far" observer at cell number 2700. Fig. 4-1 shows the numerical experiment results as circles found using the matching method, which almost precisely agree with the theoretical curves graphed using Eqn. (4-17). Numerical experiment results from N=50 CPW to N=100 CPW for s=1, 5, and 10 can be found in [160].

4.4 Numerical Dissipation and Dispersion of 2D ADI-FDTD

The 2D ADI-FDTD can be formulated by expanding Eqn. (4-12), which is the same as Reference [81]. The update equations for 2D ADI-FDTD are as follows.

1st sub-step	2nd sub-step	
$(1-b^2D_{2x})E_y^{n+1/2}$	$(1-b^2D_{2y})E_x^{n+1}$	
$=E_y^n-a_1D_xH_z^n-b^2D_xD_yE_x^n$	$=E_x^{n+1/2}+a_1D_yH_z^{n+1/2}-b^2D_xD_yE_y^{n+1/2}$	(4-18a)
$E_x^{n+1/2}=E_x^n+a_1D_yH_z^n$	$E_y^{n+1}=E_y^{n+1/2}-a_1D_xH_z^{n+1/2}$	(4-18b)
$H_z^{n+1}=H_z^n+a_2D_yE_x^n-a_2D_xE_y^{n+1/2}$	$H_z^{n+1}=H_z^{n+1/2}+a_2D_yE_x^{n+1}-a_2D_xE_y^{n+1/2}$	(4-18c)

With the Fourier analysis, the amplification polynomials can be obtained as Eqn. (4-19), and the amplification factors for the two sub-marching procedures can be found and rewritten in amplitude and phase as [152]

$$P(\xi_1)=(1-\xi_1)\left(1+r_y^2-2\xi_1+(1+r_x^2)\xi_1^2\right) \tag{4-19a}$$

$$P(\xi_2)=(1-\xi_2)\left(1+r_x^2-2\xi_2+(1+r_y^2)\xi_2^2\right) \tag{4-19b}$$

$$\xi_1=\xi_{10}e^{\pm j\varphi_1} \qquad \xi_2=\xi_{20}e^{\pm j\varphi_2} \tag{4-20a}$$

$$\xi_{10}=\frac{1}{\xi_{20}}=\sqrt{\frac{1+r_y^2}{1+r_x^2}} \qquad \varphi_1=\varphi_2=\tan^{-1}\sqrt{(1+r_x^2)(1+r_y^2)-1} \tag{4-20b}$$

Eqn. (4-20b) shows clearly that the magnitudes of the two amplification factors are reciprocals and the phases are the same. In addition, the magnitude of the amplification factors changes with the direction of travel for a given mesh size (see Fig.1 in [152]), due to the unbalanced time-splitting in the two sub-marching procedures.

The numerical dispersion relation for the 2D ADI-FDTD method is derived as [152]

$$\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}=\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+(c\Delta t)^2\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} \qquad (4\text{-}21)$$

It can be shown [152] that Eqn. (4-21) is equivalent to the numerical dispersion relation in [99]. The inconsistency concerning the numerical dispersion relation among [81], [99] and [100] is due to the fact that Reference [81] and [100] assume that the magnitudes of the two amplification factors are unity, which is incorrect.

Fig.4-2 shows the relative wave velocity as a function of the direction of travel for Courant number $s = c\Delta t / \Delta x$ equal to 1, 5 and 10 calculated with Eqn. (4-21). Fig.4-2 also compares the theoretical prediction (solid line) from Eqn. (4-21) with the results from numerical experiment (small circles). It can be seen that they agree well.

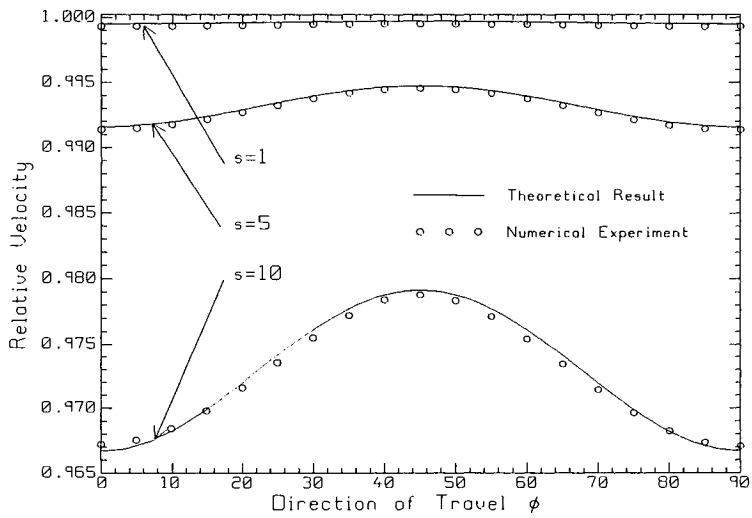

Fig. 4-2 Numerical dispersion at mesh density 100 in 2D.

Notice that the numerical anisotropy is large, and it increases as the Courant number increases. To take into account the case that the numerical velocity along axis u_a may be larger than that along the diagonals u_d, the anisotropy definition in Eqn. (3-1) is used [153]. This definition has the same magnitude as that in [4], but can be positive or negative, depending on which velocity is larger. Fig.4-3 shows the numerical anisotropy for ADI, which is much larger than CN and CNDG [153]. For a Courant number 10 at mesh density 50 CPWs, the anisotropy is 0.058 for ADI, and 0.055×0.01 for CN; at mesh density 100 CPWs, the anisotropy is 0.013 for ADI, and 0.013×0.01 for CN. Thus ADI has two-order of magnitude larger anisotropy than the CN scheme. Compared to the CN scheme Eqn. (4-5), the last term in Eqn. (4-21) is extra, which is responsible for large anisotropy.

However, ADI has a smaller numerical dispersion error along diagonals than the CN scheme, though the numerical dispersion is the same along the axes, which can be seen that along the x-axis, the numerical dispersion relation becomes $\tan^2(\omega\Delta t/2) = (c\Delta t/\Delta x)^2 \sin^2(\beta_x \Delta x/2)$ for both methods. Thus, in terms numerical dispersion, ADI is better than CN; in terms of numerical anisotropy, CN (and other CN-based methods in Chapters 5 and 6) is superior to ADI. Note that if one method has smaller anisotropy, it is easy to reduce the numerical dispersion by coefficient-modification techniques discussed in Chapter 3.

Detailed comparison for ADI, CN and other CN-based methods can be found in [153]. Though at each sub-step ADI-FDTD has either growth or dissipation, the numerical dispersion relation can be separated from the dissipation because the overall ADI method is strictly non-dissipative.

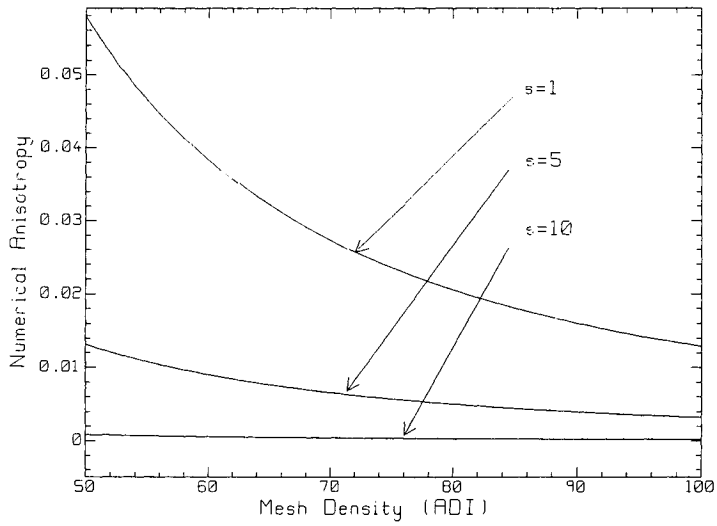

Fig. 4-3 Numerical anisotropy of 2D ADI-FDTD at the Courant numbers 1, 5 and 10.

4.5 Perturbation Errors for 2D ADI-FDTD

From previous description, it can be seen that ADI method is really a perturbation of the CN scheme [104]. Since Eqn. (4-11) has been split into two sub-steps Eqn. (4-12), to find what terms have been added to the CN scheme in formulating ADI-FDTD, the two sub-steps in ADI must be re-combined into one-step formulation. To achieve this goal, Eqn. (4-12) can be re-written as

$$W^{n+1/2} = (I - D_1)^{-1}(I + D_2)W^n \tag{4-23a}$$

$$W^{n+1} = (I - D_2)^{-1}(I + D_1)W^{n+1.2} \tag{4-23b}$$

Thus the 2D ADI-FDTD method can be formulated in one-step as

$$W^{n+1} = (I - D_2)^{-1}(I + D_1)(I - D_1)^{-1}(I + D_2)W^n \tag{4-24}$$

Expanding the matrices obtains the following one-step ADI formulation

$$(1-b^2D_{2x})(1-b^2D_{2y})\begin{pmatrix}E_x^{n+1}\\E_y^{n+1}\\H_z^{n+1}\end{pmatrix}=$$

$$\begin{pmatrix}1-b^2D_{2x}+b^2D_{2y}+b^4D_{2x}D_{2y} & -2b^2D_xD_y & 2D_y\\ -2b^2D_xD_y(1-b^2D_{2y}) & 1+b^2D_{2x}-b^2D_{2y}-b^4D_{2x}D_{2y} & -2D_x(1-b^2D_{2y})\\ 2D_y & -2D_x & 1+b^2D_{2x}+b^2D_{2y}-b^4D_{2x}D_{2y}\end{pmatrix}\begin{pmatrix}E_x^n\\E_y^n\\H_z^n\end{pmatrix} \tag{4-24}$$

Writing Eqn. (4-24) in individual update equations reads

$$\{1-b^2(D_{2x}+D_{2y})\}E_x^{n+1}=\{1-b^2(D_{2x}-D_{2y})\}E_x^n-2b^2D_xD_yE_y^n+2a_1D_yH_z^n \\ -b^4D_{2x}D_{2y}(E_x^{n+1}-E_x^n) \tag{4-25a}$$

$$\{1-b^2(D_{2x}+D_{2y})\}E_y^{n+1}=\{1+b^2(D_{2x}-D_{2y})\}E_y^n-2b^2D_xD_yE_x^n-2a_1D_xH_z^n \\ -b^4D_{2x}D_{2y}(E_y^{n+1}+E_y^n)+2b^4D_xD_{3y}E_y^n+2b^3D_xD_{2y}H_z^n \tag{4-25b}$$

$$\{1-b^2(D_{2x}+D_{2y})\}H_z^{n+1}=\{1+b^2(D_{2x}+D_{2y})\}H_z^n+2a_2(D_yE_x^n-D_xE_y^n) \\ -b^4D_{2x}D_{2y}(H_z^{n+1}+H_z^{n+1}) \tag{4-25c}$$

Compared to the CN scheme of Eqs. (4-2d), (4-2e) and (4-2a), the terms in the second line for each equations in (4-25) are the extra terms added to formulate ADI. Dividing by Δt in both sides of Eqn. (4-25) one can find the order of accuracy. Those terms are $-b^4D_{2x}D_{2y}(E_x^{n+1}-E_x^n)/\Delta t$ in Eqn. (4-25a), which is 4th order accurate in time, $\left(-b^4D_{2x}D_{2y}(E_y^{n+1}+E_y^n)+2b^4D_xD_{3y}E_x^n+2b^3D_xD_{2y}H_z^n\right)/\Delta t$ in Eqn. (4-15b), which are 4th order, 3rd order and 2nd order in time, and $-D_{2x}D_{2y}(H_z^{n+1}+H_z^n)/\Delta t$ in Eqn. (4-15c) is 4th order in time. They are the perturbation errors. Because the electric and the magnetic fields are coupled with each other, one may not conclude that E_y has the poorest accuracy because of the 2nd order perturbation. It can be also seen that the perturbation errors are different from the factorization error, possibly because of splitting and the intermediate time step introduced.

Reference [104] points out that ADI-FDTD is an $O(\Delta t^2)$ perturbation to CN, and the truncation error grows with Δt^2 and only gives the 2^{nd} –order accurate term. The above derivation confirms this conclusion, and gives explicitly all the terms of perturbations.

4.6 Isotropic ADI-FDTD

ADI-FDTD can use a larger time step size than the Courant limit. However, the sacrifice is larger numerical dispersion error than Yee's FDTD, and larger numerical anisotropy, which is two orders-of-magnitude larger than the CN scheme [153]. Higher-order ADI-FDTD [94-97] can reduce the error, but higher than 6^{th} order provides little improvement for large Courant numbers [95]. Reference [84] comments that high-order does not improve as much as expected. Our analysis shows that the higher-order (2,4) method "over-corrects" the dispersion error, and anisotropy still exists though becomes smaller. References [38] [98] propose an anisotropy method that can eliminate errors at only two directions of travel between $\phi = 0°$ and $\phi = 90°$ for a non-uniform mesh, and along the axes for a uniform mesh. However, the inherent anisotropy is still larger than CN-based methods.

Since ADI is a time-stepping scheme, it has no special requirement for the discretization of the spatial derivatives. Therefore the O24 method proposed in Chapter 3 for explicit FDTD can be used to eliminate numerical dispersion along the diagonals, or along the axes, or eliminate anisotropy for a uniform mesh. By the use of the "isotropic-optimization" formulation discussed in Chapter 3, a new method can be created, termed "isotropic ADI" (IADI). The numerical dispersion relation for IADI-FDTD can be derived as

$$\tan^2(\omega\Delta t/2)/(c\Delta t)^2 = r_x^2(w+(1-w)p_{x24})^2 + r_y^2(w+(1-w)p_{y24})^2 + r_x^2 r_y^2(w+(1-w)p_{x24})^2(w+(1-w)p_{y24})^2 \quad (4\text{-}26)$$

To obtain the optimal value of the weight parameter w with isotropic velocity, the following equation must be solved

$$r_{x,0}^2(w+(1-w)p_{x24,0})^2 = 2r_{x,45}^2(w+(1-w)p_{x24,45})^2 + r_{x,45}^4(w+(1-w)p_{x24,45})^4 \quad (4\text{-}27)$$

However, the above equation cannot be solved analytically. The optimal value of the weight parameter w can be found using the same technique of search introduced in Chapter 3. Note that in the ADI case, the value of the optimized weight parameter depends on not only the mesh density but also on the time step size. Fig.4-4 shows the numerical dispersion using the IADI-FDTD method. For comparison, the numerical dispersion for

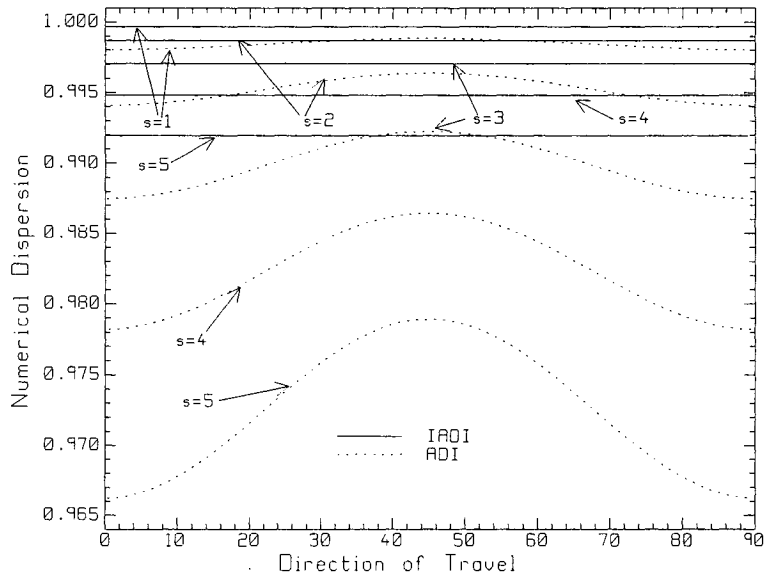

Fig. 4-4 The numerical dispersion of IADI-FDTD and ADI-FDTD at mesh density 50 CPWs and Courant number 1, 2, 3, 4 and 5.

the original ADI-FDTD is shown in the same figure. It can be seen that the anisotropy is indeed suppressed for the IADI-FDTD method, and the (2,4) stencil gives much better compensation to the numerical dispersion. Different from explicit case, the resulting numerical velocity is smaller than the theoretical speed. This is may be because of large numerical dispersion along the axes for the ADI method. In the ADI case, the one-cell-away formulation "under-corrects" the dispersion. The under-corrected dispersion can be eliminated by modifying the coefficients of the update equation using the technique proposed in Chapter 3, where the parameter optimized method [38] only eliminates the numerical dispersion along some specific directions of travel.

4.7 3D ADI-FDTD Formulation

It has been noticed that, in 2D ADI, the matrix D_1 contains all the difference operators having negative sign, and D_2 contains the difference operators having positive sign in the RHS of the Curl equations. 3D ADI-FDTD can also be formulated similarly. The matrices are

$$D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_1 D_z & 0 \\ 0 & 0 & 0 & 0 & 0 & -a_1 D_x \\ 0 & 0 & 0 & -a_1 D_y & 0 & 0 \\ 0 & 0 & -a_2 D_y & 0 & 0 & 0 \\ -a_2 D_z & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_2 D_x & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4-28a}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & a_1 D_y \\ 0 & 0 & 0 & a_1 D_z & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 D_x & 0 \\ 0 & a_2 D_z & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 D_x & 0 & 0 & 0 \\ a_2 D_y & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4-28b}$$

Eqs. (4-11) and (4-12) are still valid in the 3D case. To formulate Namaki's 3D ADI-FDTD [82], directly use the above matrices; to formulate Zheng-Chen-Zhang's 3D ADI-FDTD [4] [83], just exchange the position of D_1 and D_2. The numerical dispersion relation is [99] [101]

$$\begin{aligned}&\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}\left(1+(c\Delta t)^6\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\right)\\&=\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+(c\Delta t)^2\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\\&+(c\Delta t)^2\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}+(c\Delta t)^2\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\end{aligned} \tag{4-29}$$

4.8 Discussion

The original formulation of ADI-FDTD [81-83] is straightforward from the discretized Maxwell's Equations when the RHS is approximated in time. This Chapter formulates the ADI-FDTD based on the CN scheme in a compact matrix form Eqs. (4-10) and (4-11) with factorization. Though the CN-based methods introduced in Chapters 5 and 6 are also formulated with the same matrix form and factorization, it is the different splitting schemes that differentiate them, and thus the numerical dispersion behavior. The factorization errors indicate why there is a cross term in the numerical dispersion relation.

It is noticed that, for current or voltage sources, ADI-FDTD must treat the source at each time step. The formulation is

$$(I-D_1)W^{n+1/2}=(I+D_2)W^n+\frac{1}{2}W_s^{n+1/2} \tag{4-30a}$$

$$(I-D_2)W^{n+1}=(I+D_1)W^{n+1/2}+\frac{1}{2}W_s^{n+1/2} \tag{4-30b}$$

where $W_s^{n+1/2}$ is the source vector evaluated at time step n+1/2. However, for CN-based methods, the source treatment is required only for the first sub-step, which is more efficient than ADI-FDTD.

Using the coefficient-modification technique, the numerical dispersion error can be minimized. The CN scheme can be re-written as

$$E_x^{n+1} = E_x^n + a_{ye} a_1 D_y (H_z^{n+1} + H_z^n) \tag{4-31a}$$

$$E_y^{n+1} = E_y^n - a_{xe} a_1 D_x (H_z^{n+1} + H_z^n) \tag{4-31b}$$

$$H_z^{n+1} = H_z^n + a_{yh} a_2 D_y (E_x^{n+1} + E_x^n) - a_{xh} a_2 D_x (E_y^{n+1} + E_y^n) \tag{4-31c}$$

The numerical dispersion is

$$\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2} = a_{xe}a_{xh}\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2} + a_{ye}a_{yh}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} + a_{xe}a_{ye}a_{xh}a_{yh}(c\Delta t)^2\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} \tag{4-32}$$

For a uniform mesh $\Delta x = \Delta y$, choosing $a_{xe} = a_{ye} = a_{xh} = a_{yh} = a_{cm}$ properly can eliminate numerical dispersion along a designated direction of travel. For example, to eliminate axial numerical dispersion, the optimized a_{cm} value will be

$$a_{cm} = \frac{\tan(\omega\Delta t/2)}{s\sin(\beta_0\Delta x/2)} \tag{4-33}$$

For other directions of travel where $r_x r_y \neq 0$, the optimized a_{cm} value can be solved from Eqn. (4-32) as

$$a_{cm} = \sqrt{\frac{\sqrt{(r_x^2 + r_y^2)^2 + 4r_x^2 r_y^2 \tan^2(\omega\Delta t/2)} - (r_x^2 + r_y^2)}{2r_x^2 r_y^2}} \tag{4-34}$$

In particular, choosing $\phi = 45°$ can eliminate dispersion error along diagonals, and choosing $\phi = 22.5°$ or $\phi = 67.5°$ can have minimum average dispersion error. Though

References [38] [98] use similar techniques to correct the dispersion, they did not give an analytical expression of a_{cm}.

4.9 Summary

ADI-FDTD is a two-step marching method, and is a 2nd order accurate perturbation to the CN scheme. At each sub-marching procedure, either numerical dissipation or numerical growth appears, because of the unbalanced formulation in time. However, this unbalanced behavior is compensated with the two sub-marching procedures, the overall method is strictly non-dissipative, and unconditionally stable. In implementation, the two sub-marching procedures must be completely fulfilled [154].

The numerical dispersion relations in 1D and 2D are re-derived using the Fourier method for each sub-step, and numerically verified with good agreement between numerical experiments and theoretical prediction. The inconsistency regarding the numerical dispersion relation is removed. Though ADI-FDTD is unconditionally stable, use of a large time step size leads to large numerical dispersion and anisotropy. The isotropic ADI-FDTD using a larger computational stencil gives much better improvement in numerical dispersion error compared to the explicit IOM method discussed in Chapter 3, and can eliminate anisotropy at one designated frequency.

It has been shown that CN-based methods [151] [157] can greatly reduce the anisotropy without using a larger computational stencil. In the next two chapters, 2D and 3D CN-based methods are described and the perturbation errors are given. These perturbation errors are 4th-order accurate in time. Thus the CN-based methods inherit the small anisotropy behavior of the original scheme at relatively small Courant numbers.

Chapter 5 2D Crank-Nicolson-Based Methods

From Chapter 4 it can be seen that ADI-FDTD can be considered as a 2nd order perturbation to the CN scheme with large anisotropy. However, ADI-FDTD differs from the CN scheme because it associates a specific time step n+1/2 with the intermediate field values, whereas the CN scheme and the CN-based methods do not associate the intermediate field values with a specific time step [151], [156-157], [163]. This thesis will classify them as the ADI methods and the CN-based methods. The CN-based methods are different from ADI also in the count of floating-point-arithmetic operations although some CN-based methods have the same numerical dispersion as the ADI methods [156]. In this chapter, the amplification polynomial, the amplification factor, the numerical dispersion relation and the numerical validation results will be given for each proposed method. The proposed methods either have higher computational efficiency, or have smaller anisotropy than ADI-FDTD.

5.1 An Efficient Splitting Scheme

Eqn (4-11) can be split into two steps in various ways, such as the D'Yakonov scheme, Douglas-Rechford scheme, Peaceman-Rechford scheme, and the Beam-Warming scheme [76-78]. However, different splitting schemes may lead to quite different numerical accuracy and computational stencils, particularly for non-homogenous difference equations such as

$$(I-D_1)(I-D_2)W^{n+1}=(I+D_1)(I+D_2)W^n+f \tag{5-1}$$

where f is a matrix, which can be a current or voltage source vector in Eqn. (4-30), or other known terms introduced in this Chapter. To avoid numerical dissipation and to alleviate the direct computation of $D_1D_2W^n$ which requires a large computational

stencils, and to have fewer floating-point-arithmetic operations, this thesis proposes an "efficient splitting" scheme with intermediate values W^* as

$$(I-D_1)W^* = (I+D_1+2D_2)W^n + f \tag{5-2a}$$

$$(I-D_2)W^{n+1} = W^* - D_2 W^n \tag{5-2b}$$

It will be shown that by numerical dispersion analysis and numerical experiments that such an efficient splitting scheme performs well. Compared to Eqn. (4-30), the matrix f is treated only once instead of twice in the ADI formulation Eqn. (4-30). This efficient-splitting scheme will be used in efficiently developing CN-based schemes in this thesis. Note that dropping the cross term $D_1 D_2$ in RHS of Eqn. (5-1), as done by some authors in order to reduce the computational stencils, will introduce artificial numerical attenuation, which can be avoided by Eqn. (5-2).

5.2 A Direct-Splitting Method

Using the efficient splitting scheme Eqs. (5-2) and (4-10), the Eqn. (4-11) can be written in the form

$$E_x^* = E_x^n + 2a_1 D_y H_z^n \tag{5-3a}$$

$$E_y^* + a_1 D_x H_z^* = E_y^n - a_1 D_x H_z^n \tag{5-3b}$$

$$H_z^* + a_2 D_x E_y^* = H_z^n + 2a_2 D_y E_x^n - a_2 D_x E_y^n \tag{5-3c}$$

$$E_x^{n+1} - a_1 D_y H_z^{n+1} = E_x^* - a_1 D_y H_z^n \tag{5-3d}$$

$$E_y^{n+1} = E_y^* \tag{5-3e}$$

$$H_z^{n+1} - a_2 D_y E_x^{n+1} = H_z^* - a_2 D_y E_x^n \tag{5-3f}$$

and will be termed as the Crank-Nicolson-direct-splitting (CNDS) method. Further simplification to Eqn. (5-3) shows that similar to ADI, CNDS solves two implicit tridiagonal equations: one for the intermediate time step, and the other for the current time step. The rest of the field components can be updated explicitly. Note that Eqn. (5-3)

can be written much more concisely as the Crank-Nicolson-approximate-decoupling E_y(CNAD-Y) method [156] presented in Section 5.4. Exchanging D_1 and D_2 can lead to a different formulation. The former can be called CNDS-Y, the later can be called CNDS-X to distinguish the different formulation. As pointed out in Chapter 4, the factorization error CNDS-Y method is $-b^2 D_x D_y (E_x^{n+1} - E_x^n)$ to the E_y update equation, which is the perturbation error to the original CN scheme.

Note from Eqn. (5-3) it can be seen that the CNDS methods can also be formulated by solving the magnetic field component implicitly and then the electric field components explicitly. With some manipulation, the implicit update equations are

$$H_z^* - b^2 D_{2x} H_z^* = H_z^n + b^2 D_{2x} H_z^n + 2a_2 D_y E_x^n - 2a_2 D_x E_y^n \tag{5-4a}$$

$$H_z^{n+1} - b^2 D_{2y} H_z^{n+1} = H_z^* - b^2 D_{2y} H_z^n \tag{5-4b}$$

This can be termed as CNDS-H method.

Using the Fourier analysis method obtains the amplification polynomial for all the CNDS methods as

$$P(\xi) = (1-\xi)\left((1+r_x^2)(1+r_y^2)(\xi+1)^2 - 4\xi\right) \tag{5-5a}$$

The amplification factor of the propagating mode and the numerical dispersion relation can be derived as

$$\xi = \frac{1 - r_x^2 - r_y^2 - r_x^2 r_y^2 \pm J\sqrt{(1+r_x^2)(1+r_y^2)^2 - (1 - r_x^2 - r_y^2 - r_x^2 r_y^2)^2}}{(1+r_x^2)(1+r_y^2)} \tag{5-5b}$$

$$\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2} = \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2} + \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} + (c\Delta t)^2 \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} \tag{5-5c}$$

The amplification factor has a unity magnitude, thus CNDS method is unconditionally stable. The numerical dispersion relation Eqn. (5-5c) is the same as ADI-FDTD Eqn. (4-

21) [152], thus the CNDS has larger anisotropy errors than the CN scheme. In the next few sections some more accurate methods for efficiently implementing the CN scheme will be discussed.

5.3 The Crank-Nicolson-Douglas-Gunn Method

From Eqn. (4-1) it can be seen that the discrete electric and magnetic field components are coupled, which lead to a huge sparse matrix to be solved expensively. One way to decouple the electric and magnetic fields is to insert Eqs. (4-1a) and (4-1b) into (4-1c) to eliminate the implicit electric field components at the current time step n+1 [151]. With some manipulation, it can be written as

$$\left(1-b^2(D_{2x}+D_{2y})\right)H_z^{n+1}=\left(1+b^2(D_{2x}+D_{2y})\right)H_z^n+2a_2\left(D_yE_x^n-D_xE_y^n\right) \tag{5-6a}$$

Note Eqn. (5-5) leads to a block tridiagonal matrix [77], which is expensive to solve efficiently. By adding terms $b^4D_{2x}D_{2y}H_z^{n+1}$ and $b^4D_{2x}D_{2y}H_z^n$ to the LHS and RHS of Eqn. (5-6a), respectively, it can be factorized into

$$\left(1-b^2D_{2x}\right)\left(1-b^2D_{2y}\right)H_z^{n+1}=\left(1+b^2D_{2x}\right)\left(1+b^2D_{2y}\right)H_z^n+2a_2\left(D_yE_x^n-D_xE_y^n\right) \tag{5-6b}$$

This implicit equation can be solved by the Douglas-Gunn method [77] [151] (Jim Douglas, Jr., 1927-, James E. Gunn, 1938-). In [151] large computational stencils are encountered since the cross-derivative term $D_{2x}D_{2y}$ is left in the RHS. By the use of the efficient splitting method, Eqn. (5-6b) can be solved with the following two-step update equations

$$\left(1-b^2D_{2x}\right)H_z^*=\left(1+b^2D_{2x}+2b^2D_{2y}\right)H_z^n+2a_2\left(D_yE_x^n-D_xE_y^n\right) \tag{5-7a}$$

$$\left(1-b^2D_{2y}\right)H_z^{n+1}=H_z^*-b^2D_{2y}H_z^n \tag{5-7b}$$

This formulation avoids large computational stencils [151], and is still termed as the Crank-Nicolson-Douglas-Gunn (CNDG) method. The amplification coefficient matrix,

the amplification polynomial and factor as well as the numerical dispersion relation can be derived by the use of the Fourier method as

$$Q=\begin{pmatrix} 1-\xi & 0 & -J2a_1(1+\xi)\sin(\beta_y\Delta y/2)/\Delta y \\ 0 & 1-\xi & J2a_1(1+\xi)\sin(\beta_x\Delta x/2)/\Delta x \\ -J4a_2\sin(\beta_y\Delta y/2)/\Delta y & J4a_2\sin(\beta_x\Delta x/2)/\Delta x & (1-r_x^2)(1-r_y^2)-(1+r_x^2)(1+r_y^2)\xi \end{pmatrix} \tag{5-8a}$$

$$P(\xi)=(1-\xi)\left((1-\xi)\left((1-r_x^2)(1-r_y^2)-\xi(1+r_x^2)(1+r_y^2)\right)+2(r_x^2+r_y^2)(\xi+1)\right) \tag{5-8b}$$

$$\xi=\frac{\left((1-r_x^2)(1-r_y^2)\right)\pm J\sqrt{\left((1+r_x^2)(1+r_y^2)\right)^2-\left((1-r_x^2)(1-r_y^2)\right)^2}}{\left((1+r_x^2)(1+r_y^2)\right)}=e^{\pm J\tan^{-1}\frac{\sqrt{\left((1+r_x^2)(1+r_y^2)\right)^2-\left((1-r_x^2)(1-r_y^2)\right)^2}}{\left((1-r_x^2)(1-r_y^2)\right)}} \tag{5-8c}$$

$$\left\{1+(c\Delta t)^4\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\right\}\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}=\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} \tag{5-8d}$$

The CNDG method has a unity-magnitude amplification factor, thus it is unconditionally stable. Numerical experiments and analysis [151] [153] show that CNDG has much smaller anisotropy than ADI and CNAD, and has the fewest floating-point-arithmetic counts (see Section 5.7.5).

5.4 The Coupled Electric Field Components at the Current Time Step

In many practical problems, perfect electric conductors (PECs) are often encountered. Solving the problem in term of the electric field components makes the enforcement of the boundary conditions easier than solving in terms of the magnetic field. To this end, inserting Eqn. (4-1c) into Eqs. (4-1a) and (4-1b), respectively, obtains the following equations

$$(1-b^2D_{2y})E_x^{n+1}+b^2D_xD_yE_y^{n+1}=(1+b^2D_{2y})E_x^n-b^2D_xD_yE_y^n+2a_1D_yH_z^n \tag{5-9a}$$

$$(1-b^2D_{2x})E_y^{n+1}+b^2D_xD_yE_x^{n+1}=(1+b^2D_{2x})E_y^n-b^2D_xD_yE_x^n-2a_1D_xH_z^n \tag{5-9b}$$

Eqn. (5-9) realizes the decoupling between the magnetic and the electric fields at the current time step n+1, but the two electric field components are still coupled each other. Directly solving Eqn. (5-9) is still expensive in terms of computer resources. However, it

can be solved efficiently by adding some 2nd order- or higher-order terms inspired by the ADI method. With approximate factorization and the efficient splitting method, Eqn. (5-9) can be solved efficiently from the following two-step equations

$$(I-A)W^{*} = (I+A+2B)W^{n} + f_e \tag{5-10a}$$

$$(I-B)W^{n+1} = W^{*} - BW^{n} \tag{5-10b}$$

where now $W^n = \begin{pmatrix} E_x^n & E_y^n \end{pmatrix}^T$, $f_e = 2a_1 \begin{pmatrix} D_y & -D_x \end{pmatrix}^T H_z^n$, and A and B are 2 by 2 difference matrices. Depending on the formulation of A and B, several different methods can be devised [156] as demonstrated in the following. It will be shown later that Eqn. (5-10) has been added some 4th order-accurate terms implicitly.

5.5 Crank-Nicolson Approximate-Decoupling Method

Observing Eqn. (5-9), the coupling between E_x^{n+1} and E_y^{n+1} is due to the $D_x D_y$ term. When solving for E_y^{n+1} implicitly first, the 2nd order terms $-D_x D_y E_x^{n+1}$ and $-D_x D_y E_x^n$ can be added to the LHS and to the RHS of Eqn. (5-9b), respectively. Thus the decoupling is realized. This method is termed as Crank-Nicolson-approximate-decoupling E_y (CNAD-Y) method. The update equations are

$$(1-b^2 D_{2x}) E_y^{n+1} = (1+b^2 D_{2x}) E_y^n - 2a_1 D_x H_z^n - 2b^2 D_x D_y E_x^n \tag{5-11a}$$

$$(1-b^2 D_{2y}) E_x^{n+1} = (1+b^2 D_{2y}) E_x^n + 2a_1 D_y H_z^n - b^2 D_x D_y (E_y^{n+1} + E_y^n) \tag{5-11b}$$

$$H_z^{n+1} = H_z^n + a_2 D_y (E_x^{n+1} + E_x^n) - a_2 D_x (E_y^{n+1} + E_y^n) \tag{5-11c}$$

Similarly, by adding 2nd order terms $-b^2 D_x D_y (E_y^{n+1} - E_y^n)$ to Eqn. (5-9a), Eqn. (5-9) is solved by CNAD-X method [156] without the intermediate value. This CNAD method solves two tridiagonal matrices implicitly, and updates the magnetic field Eqn. (5-11c) explicitly, at each time step. Note that CNDS-Y has the same update equations as Eqn. (5-11) after some manipulation. The amplification coefficient matrix is

$$Q=\begin{pmatrix} 1-r_y^2-(1+r_y^2)\xi & r_x r_y(1+\xi) & -J4a_1\sin(\beta_y\Delta y/2)/\Delta y \\ 2r_x r_y & 1-r_x^2-(1+r_x^2)\xi & J4a_1\sin(\beta_x\Delta x/2)/\Delta x \\ -J2a_2(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y & J2a_2(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x & 1-\xi \end{pmatrix} \tag{5-12}$$

From the determinant of Q, the amplification polynomial can be obtained as

$$P(\xi)=(\xi-1)\left((1+r_x^2)(1+r_y^2)(\xi^2+1)+2(r_x^2r_y^2-r_x^2-r_y^2-1)\xi\right) \tag{5-13}$$

The amplification factor and the numerical dispersion relation are the same as Eqn. (5-5). Note that adding terms $-D_xD_yE_x^{n+1}$ and $-D_xD_yE_x^n$ into Eqn. (5-9a) allows solving for E_x^{n+1} implicitly first [156]

$$(1-b^2D_{2y})E_x^{n+1}=E_x^n+2a_1D_yH_z^n-2b^2D_xD_yE_y^n+b^2D_{2y}E_x^n \tag{5-14a}$$

$$(1-b^2D_{2x})E_y^{n+1}=E_y^n-2a_1D_xH_z^n+D_{2x}E_y^n-D_xD_yE_x^n-D_xD_yE_x^{n+1} \tag{5-14b}$$

This method decouples E_x first, thus is termed as CNAD-X. Therefore the CNAD method can be formulated with two different algorithms Eqs. (5-11) and (5-14). Both algorithms have the same amplification factor and thus the same numerical dispersion relation Eqn. (5-5c), although their amplification coefficient matrices are different.

5.6 Crank-Nicolson Cycle-Sweep Method

The cycle-sweep method solves Eqn. (5-9) by use of different formulations of the matrices A and B, and four algorithms were presented in [156]. They can be solved either first explicitly, then implicitly, or first implicitly and then explicitly.

5.6.1 Crank-Nicolson-cycle-sweep-implicit-explicit method

By constructing the matrices A and B as follows

$$A=b^2\begin{bmatrix} D_{2y} & 0 \\ -D_xD_y & D_{2x} \end{bmatrix} \qquad B=b^2\begin{bmatrix} 0 & -D_xD_y \\ 0 & 0 \end{bmatrix} \tag{5-15}$$

the following update equations can be obtained from Eqn. (5-10)

$$(1-b^2D_{2y})E_x^* = E_x^n + 2a_1D_yH_z^n - 2b^2D_xD_yE_y^n + b^2D_{2y}E_x^n \tag{5-16a}$$

$$(1-b^2D_{2x})E_y^{n+1} = E_y^n - 2a_1D_xH_z^n + D_{2x}E_y^n - D_xD_yE_x^n - D_xD_yE_x^* \tag{5-16b}$$

$$E_x^{n+1} = E_x^* + b^2D_xD_yE_y^n - b^2D_xD_yE_y^{n+1} \tag{5-16c}$$

This method solves for E_x^* and E_y^{n+1} implicitly with tridiagonal matrices, and then solves E_x^{n+1} explicitly in a "cycle-sweep" manner: it begins with and ends with E_x, thus it is termed the Crank-Nicolson "cycle-sweep implicit-explicit" (CNCS-IEX) algorithm.

Notice that Eqs. (5-16a) and (5-16b) are the same as Eqn. (5-14) of the CNAD method. Eqn. (5-16c) compensates explicitly for the terms $D_xD_y(E_y^{n+1} - E_y^n)$ dropped out in Eqs. (5-14a) and (5-16a). This compensation does make a difference for the amplification factor and thus for the numerical dispersion. The amplification coefficient matrix, the amplification factor, the amplification polynomial and the numerical dispersion relation can be obtained as

$$Q=\begin{pmatrix} 1-r_y^2-(1+r_y^2)\xi & r_xr_y(1-r_y^2+(1+r_y^2)\xi) & -J4a_1\sin(\beta_y\Delta y/2)/\Delta y \\ r_xr_y(1+\xi) & 1-r_x^2+r_x^2r_y^2-(1+r_x^2+r_x^2r_y^2)\xi & J4a_1\sin(\beta_x\Delta x/2)/\Delta x \\ -J2a_2(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y & J2a_2(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x & 1-\xi \end{pmatrix} \tag{5-17a}$$

$$P(\xi) = \left(1+r_x^2\right)\left(1+r_y^2\right)\left(\xi^3-1\right)-\left(3-r_x^2-r_y^2+3r_x^2r_y^2\right)\left(\xi-1\right) \tag{5-17b}$$

$$\xi = \frac{\left((1-r_x^2)(1-r_y^2)\right)\pm J\sqrt{\left((1+r_x^2)(1+r_y^2)\right)^2-\left((1-r_x^2)(1-r_y^2)\right)^2}}{\left((1+r_x^2)(1+r_y^2)\right)} \tag{5-17c}$$

$$\left\{1+(c\Delta t)^4\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\right\}\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2} = \frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} \tag{5-17d}$$

It can be seen from Eqn.(5-17c) that the CNCS-IEX method is unconditionally stable and strictly non-dissipative. Comparing the numerical dispersion relation Eqn. (5-17d) to Eqn. (5-8b) for CNAD, their difference is the location of the cross term

$\left(\tan^2(\omega\Delta t/2)/(c\Delta t)^2\right)\left(\sin^2(\beta_x\Delta x/2)/\Delta x^2\right)\left(\sin^2(\beta_y\Delta y/2)/\Delta y^2\right)$: when it appears in the RHS it will result in larger anisotropy [156].

Similarly CNCS-IEX) algorithm can be constructed. It begins the sweep with and ends with E_y, with the following matrices

$$A = b^2\begin{bmatrix} D_{2y} & -D_xD_y \\ 0 & D_{2x} \end{bmatrix} \qquad B = b^2\begin{bmatrix} 0 & 0 \\ -D_xD_y & 0 \end{bmatrix} \tag{5-18}$$

The update equations are

$$(1-b^2D_{2x})E_y^* = E_y^n - 2a_1D_xH_z^n + b^2D_{2x}E_y^n - 2D_xD_yE_x^n \tag{5-19a}$$

$$(1-b^2D_{2y})E_x^{n+1} = E_x^n + 2a_1D_yH_z^n - b^2D_xD_y(E_y^* + E_y^n) + b^2D_{2y}E_x^n \tag{5-19b}$$

$$E_y^{n+1} = E_y^* + b^2D_xD_yE_x^n - b^2D_xD_yE_x^{n+1} \tag{5-19c}$$

This method solves for E_y^* and E_x^{n+1} implicitly with tridiagonal matrices, and then solves E_y^{n+1} explicitly in the cycle-sweep manner. Thus it is termed the Crank-Nicolson "cycle-sweep implicit-explicit" (CNCS-IEY) algorithm. The amplification coefficient matrix is

$$Q=\begin{pmatrix} 1-r_y^2+r_x^2r_y^2-(1+r_y^2+r_x^2r_y^2)\xi & r_xr_y(1+\xi) & -J4a_1\sin(\beta_y\Delta y/2)/\Delta y \\ r_xr_y(1-r_x^2+(1+r_x^2)\xi) & 1-r_x^2-(1+r_x^2)\xi & J4a_1\sin(\beta_x\Delta x/2)/\Delta x \\ -J2a_2(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y & J2a_2(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x & 1-\xi \end{pmatrix} \tag{5-20}$$

It can be shown that the amplification polynomial, the amplification factor and the numerical dispersion relation are the same as Eqn. (5-18) for CNCS-IEX.

5.6.2 Crank-Nicolson-cycle-sweep-explicit-implicit method

The CNCS-IE algorithms proposed above first approximate the cross term $D_xD_yE_{x,y}^{n+1}$ with $D_xD_yE_{x,y}^n$ to realize the decoupling, and then compensate for the dropped term explicitly. This section gives another decoupling method: it drops both the second-derivatives and the cross terms, thus gives explicit update equations first. Then the final

field values are solved implicitly with the errors compensated [156]. Analysis shows that one such formulation is to construct the matrices A and B as follows

$$A = b^2\begin{bmatrix} 0 & 0 \\ -D_xD_y & 0 \end{bmatrix} \qquad B = b^2\begin{bmatrix} D_{2y} & -D_xD_y \\ 0 & D_{2x} \end{bmatrix} \tag{5-21}$$

The update equations can be formulated from Eqn. (5-10) as

$$E_x^* = E_x^n + 2a_1D_yH_z^n - 2b^2D_xD_y + 2b^2D_{2y}E_x^n \tag{5-22a}$$

$$E_y^* = E_y^n - 2a_1D_xH_z^n + 2b^2D_{2x}E_y^n - b^2D_xD_y(E_x^n + E_x^*) \tag{5-22b}$$

$$(1-b^2D_{2x})E_y^{n+1} = E_y^* - b^2D_{2x}E_y^n \tag{5-22c}$$

$$(1-b^2D_{2y})E_x^{n+1} = E_x^n - b^2D_{2y}E_x^n - b^2D_xD_y(E_y^{n+1} - E_y^n) \tag{5-22d}$$

Note that Eqn. (5-22b) can be inserted in Eqn. (5-22c) as

$$(1-b^2D_{2x})E_y^{n+1} = E_y^n - 2a_1D_xH_z^n + b^2D_{2x}E_y^n - b^2D_xD_y(E_x^n + E_x^*) \tag{5-23}$$

This algorithm solves for E_x^* explicitly, then solves E_y^{n+1} and E_x^{n+1} implicitly with two tridiagonal matrices, in the cycle-sweep manner. Thus it is termed Crank-Nicolson "cycle-sweep explicit-implicit" (CNCS-EIX) algorithm [156].

One can also construct the matrices as

$$A = b^2\begin{bmatrix} 0 & -D_xD_y \\ 0 & 0 \end{bmatrix} \qquad B = b^2\begin{bmatrix} D_{2y} & 0 \\ -D_xD_y & D_{2x} \end{bmatrix} \tag{5-24}$$

The update equations can be formulated from Eqn. (5-10) as

$$E_y^* = E_y^n - 2a_1D_xH_z^n + 2b^2D_{2x}E_y^n - 2b^2D_xD_yE_x^n \tag{5-25a}$$

$$(1-b^2D_{2y})E_x^{n+1} = E_x^n + 2a_1D_yH_z^n + b^2D_{2y}E_x^n - b^2D_xD_y(E_y^{n+1} - E_y^n) \tag{5-25b}$$

$$(1-b^2D_{2x})E_y^{n+1} = E_y^* - b^2D_{2x}E_y^n - b^2D_xD_y(E_x^{n+1} - E_x^n) \tag{5-25c}$$

which solves for E_y^* explicitly first, and then E_x^{n+1} and E_y^{n+1} implicitly with tridiagonal matrices in the cycle-sweep manner. Thus it is termed the Crank-Nicolson "cycle-sweep explicit-implicit" (CNCS-EIY) algorithm.

The amplification coefficient matrices for the CNCS-EI methods are

$$Q_{EIX}=\begin{pmatrix} 1-r_y^2-(1+r_y^2)\xi & r_x r_y(1+\xi) & -J4a_1\sin(\beta_y\Delta y/2)/\Delta y \\ r_x r_y(1-r_y^2+(1+r_y^2)\xi) & 1-r_x^2+r_x^2r_y^2-(1+r_x^2+r_x^2r_y^2)\xi & J4a_1\sin(\beta_x\Delta x/2)/\Delta x \\ -J2a_2(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y & J2a_2(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x & 1-\xi \end{pmatrix} \quad (5\text{-}26)$$

$$Q_{EIY}=\begin{pmatrix} 1-r_y^2+r_x^2r_y^2-(1+r_y^2+r_x^2r_y^2)\xi & r_x r_y(1-r_x^2+(1+r_x^2)\xi) & -J4a_1\sin(\beta_y\Delta y/2)/\Delta y \\ r_x r_y(1+\xi) & 1-r_x^2-(1+r_x^2)\xi & J4a_1\sin(\beta_x\Delta x/2)/\Delta x \\ -J2a_2(\xi+1)\sin(\beta_y\Delta y/2)/\Delta y & J2a_2(\xi+1)\sin(\beta_x\Delta x/2)/\Delta x & 1-\xi \end{pmatrix} \quad (5\text{-}27)$$

It can be shown that the amplification factors and the numerical dispersion relations are the same as for the CNCS-IE methods.

5.7 Solving the 2D Wave Equations of the Second Degree

Wave equations can often be written either as one equation of the second degree or as a system of the first degree [77]. Most books and literature focus on the numerical solution of the first-degree system. But there are some cases in which it is convenient and efficient to solve the second-degree wave equation directly. There are several explicit methods [18] [78] [127] [141-142], and References [142-144] transform the wave equation into a system of the first-degree equations. But all the explicit methods have the same Courant limit the as the first-degree system of equations. Reference [127] describes an implicit method in the 1D case, which is simple to solve because the resulting matrix is tri-diagonal. References [142-143] solve 3D Vlasov-Maxwell's Equations using ADI method. Several methods were analyzed and it was found that not all implicit methods are unconditionally stable [78]. Among the implicit methods, the Crank-Nicolson (CN) scheme is particularly preferred in this thesis because it has less anisotropy than the ADI method. This section reports a method based on the CN scheme and the Douglas-Gunn (DG) [162].

The 2D wave equation to be solved is

$$\frac{\partial^2 u}{\partial^2 t} - c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{5-28}$$

where c is the velocity, and u can be any EM field components. Since Eqn. (5-28) has a 2^{nd}-degree time derivative, it must be discretized as a three-level scheme. There are several schemes for the discretization of Eqn. (5-28). Here a scheme with small anisotropy and no numerical dissipation is desired. Using a regular rectangular mesh, one possible CN scheme can be formulated as

$$u^{n+1}(i,j)+u^{n-1}(i,j)-2u^n(i,j) = b^2(D_{2x}+D_{2y})(u^{n+1}(i,j)+u^{n-1}(i,j)+2u^n(i,j)) \tag{5-29}$$

It can be shown [162] that such a CN scheme for the 2D wave equation (5-28) has the same amplification factor and numerical dispersion relation as the first-degree Maxwell's Equations using the CN scheme, which is a reference for later modification

Eqn. (5-29) is expensive to solve. Since it is a three-level scheme, there are several methods to modify it [78] [162]. To obtain an efficient solution, first it must be approximately factorized. This can be realized by adding the higher-order terms $b^4 D_{2x}D_{2y}(u^{n+1}(i,j)+u^{n-1}(i,j))$ to the LHS and $2b^4 D_{2x}D_{2y}u^n(i,j)$ to the RHS of Eqn. (5-29), respectively, which leads to

$$(1-b^2D_{2x})(1-b^2D_{2y})(u^{n+1}(i,j)+u^{n-1}(i,j)) = 2(1+b^2D_{2x})(1+b^2D_{2y})u^n(i,j) \tag{5-30}$$

Then Eqn. (5-30) can be split for solution in various ways. With the efficient splitting method, the Douglas-Gunn algorithm [77] can be used to solve the following equation

$$(1-b^2D_{2x})(u^*(i,j)+u^{n-1}(i,j)) = 2(1+b^2D_{2x}+2b^2D_{2y})u^n(i,j) \tag{5-31a}$$

$$(1-b^2D_{2y})(u^{n+1}(i,j)+u^{n-1}(i,j)) = u^*(i,j)+u^{n-1}(i,j)-2b^2D_{2y}u^n(i,j) \tag{5-31b}$$

Such a formulation avoids large computational stencils for u^n and u^{n+1}. This method is termed as the Crank-Nicoslon-Douglas-Gunn method for the wave equation (CNDGW).

The amplification polynomial can be obtained as

$$P(\xi)=(\xi-1)\left((1+r_x^2)(1+r_y^2)(\xi^2+1)-2\xi(1-r_x^2)(1-r_y^2)\right) \tag{5-32}$$

The amplification factor is the same as the CNDG method Eqn. (5-8c) for the first–degree Maxwell's Equations [162], and thus has the same dispersion relation as Eqn. (5-8d). Therefore the CNDGW method obtains more isotropic numerical velocities than ADI-FDTD. The perturbation error is the same as its factorization error.

5.8 Numerical Validation and Discussion

To validate the methods proposed above, each algorithm was coded. In total eight programs where created. Each algorithm was run successfully for a variety of Courant numbers at mesh density from 50 to 100 CPWs in a 3000 by 3000 cell space. No instability has been found.

5.8.1 Numerical dispersion

The values of the numerical dispersion at mesh density 50 with several Courant numbers are graphed in Fig. 5-1 for CNAD and Fig. 5-2 for CNCS, respectively. The figures show excellent agreement between the theoretical dispersion values predicated by Eqs. (5-5b) and (5-20) and the numerical results. Other numerical results can be found in [151] [156] [162].

Comparing Fig. 5-1 to Fig. 5-2, it can be seen that the CNCS (CNDG) methods are much more isotropic than the CNAD method and the ADI-FDTD method (Chapter 4). Fig. 5-3 shows the anisotropy of CN and CNCS (CNDG) methods. It can be seen that the

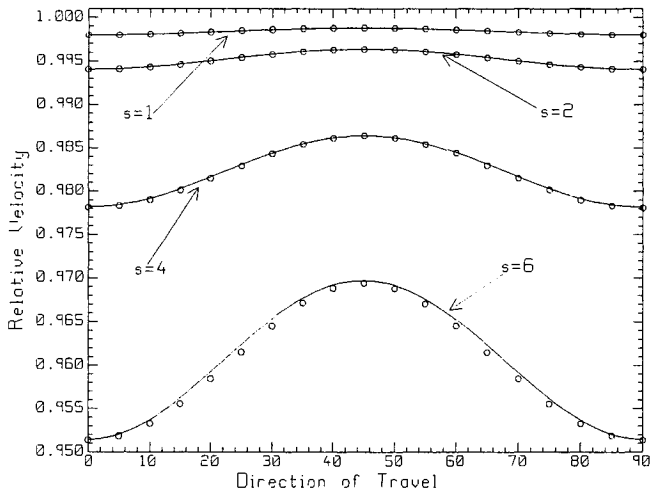

Fig. 5-1 Numerical dispersion of the CNAD method with mesh density 50. Solid line: theory; circles: numerical experiments.

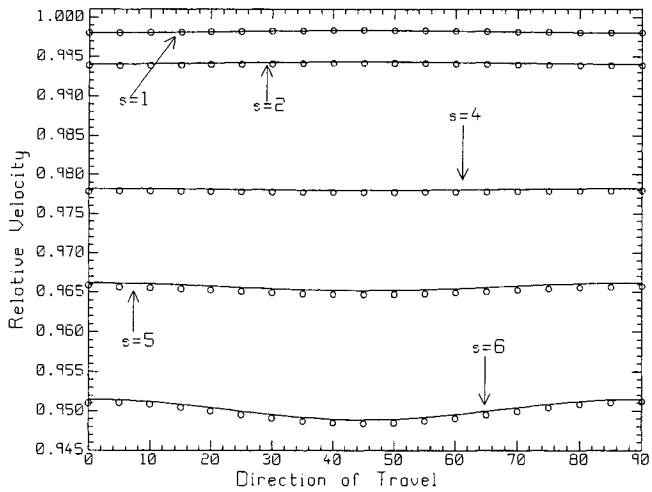

Fig. 5-2 Numerical dispersion of the CNCS method with mesh density 50. Solid line: theory; circles: numerical experiments.

anisotropy defined in Eqn. (3-1) of CNCS (CNDG) is two orders of magnitude smaller than the CNAD and ADI [153]. In Fig.5-2, at $s=1$ and $s=2$, the numerical velocity along the axes is slightly smaller than that along the diagonal $\phi=45°$, the anisotropy is positive. But for $s=4$ and $s=5$, the velocity along $\phi=45°$ is smaller than that along

axes, thus the anisotropy is negative. However, the vertical scale is Fig. 5-3 is too coarse to show the sign change of anisotropy. Such change of the anisotropy signs means that at a specific Courant number for the given mesh density, the numerical velocity is identical in all directions of travel, leading to zero anisotropy. Fig.5-4 graphs the Courant number versus the mesh density at zero anisotropy. Note that at zero anisotropy, there is numerical dispersion, but it is not large. Fig. 5-5 shows the dispersion error at zero anisotropy as a function of mesh density. At mesh densities of 50 and 100 cells, the dispersion error is 1.75% and 0.866%, respectively.

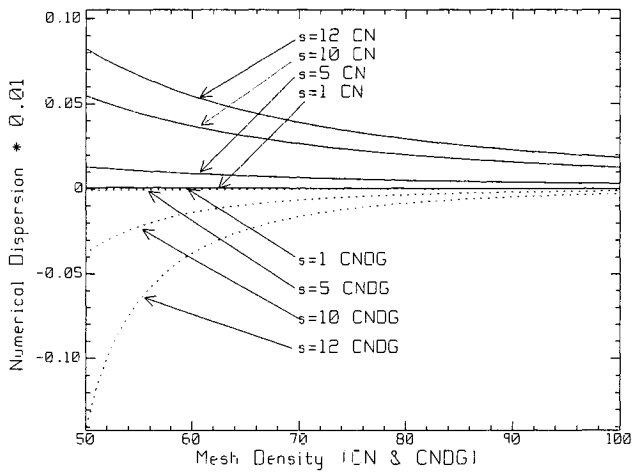

Fig. 5-3 Numerical anisotropy for CN and CNDG.

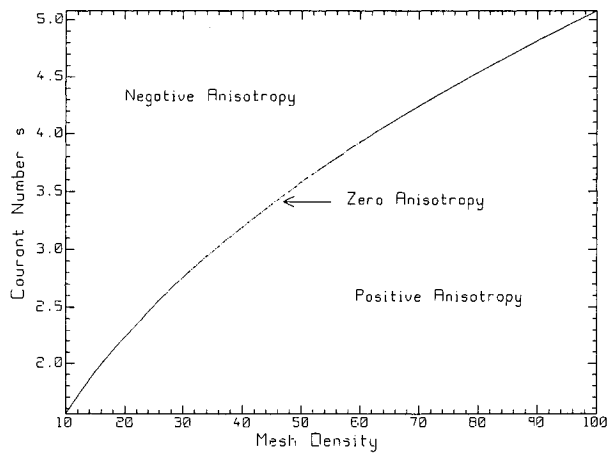

Fig. 5-4 The relation of the Courant number and mesh density at zero anisotropy for CNDG and CNCS.

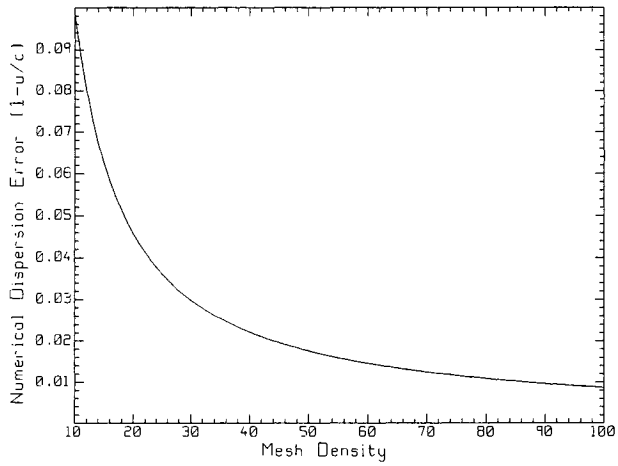

Fig. 5-5 Numerical dispersion verses mesh density at the Courant numbers without numerical anisotropy for CNCS and CNDG.

Though several methods are proposed in this chapter, the numerical dispersion relations fall into only two types: that with large anisotropy but with better numerical dispersion along the diagonals, Eqn. (5-5b), for CNAD and CNDS; and that with much less anisotropy, Eqn. (5-17d), for CNDG and CNCS. The former has a small average dispersion, the latter is suited for numerical dispersion elimination with coefficient modification, which will be discussed next.

5.8.2 Techniques to reduce the numerical dispersion

Numerical dispersion can be reduced in various ways. The simplest method is to modify the coefficients of the discretized Maxwell's Equations as discussed in Chapter 3. Since CNDS and CNAD have the same numerical dispersion relation as ADI, the discussion in Chapter 4 is applicable to CNDS and CNAD. For CNDG and CNCS, using the formulation of CN scheme with coefficient modification Eqn. (4-31), the numerical dispersion are derived as

$$\left\{1+a_{xe}a_{ye}a_{xh}a_{yh}(c\Delta t)^4\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\right\}\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}$$
$$=a_{xe}a_{xh}\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+a_{ye}a_{yh}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2} \tag{5-33}$$

For a uniform mesh $\Delta x=\Delta y$, choose $a_{xe}=a_{ye}=a_{xh}=a_{yh}=a_{cm}$. To eliminate axial numerical dispersion, the optimized a_{cm} value in Eqn. (4-33) can be used. For other directions of travel where $r_x r_y \neq 0$, the optimized a_{cm} value can be solved as

$$q=\sqrt{\frac{(r_x^2+r_y^2)-\sqrt{(r_x^2+r_y^2)^2-4r_x^2r_y^2\tan^4(\omega\Delta t/2)}}{2r_x^2r_y^2\tan^2(\omega\Delta t/2)}} \tag{5-34}$$

Fig.5-6 shows the numerical test results using the coefficient-modification technique for CNCS method (CNCS-CM) at a mesh density 50 CPW for the Courant number of 1, 2 and 4. For comparison, the numerical dispersion at Courant number 1 and 4 is also shown for CNCS without coefficient modification. It can be seen that the numerical dispersion has been almost removed with residual errors up to the accuracy of the matching method.

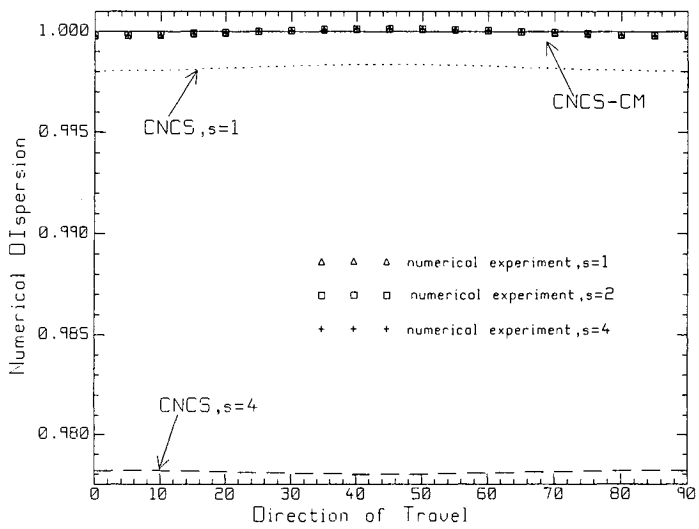

Fig. 5-6 Numerical dispersion of CNCS with the coefficient-modification technique at a mesh density 50 CPW for the Courant number of 1, 2, and 4.

5.8.3 Perturbation errors

All the methods proposed in this chapter are modifications of the CN scheme. The perturbation errors can be derived by comparing the update equations of the CN-based methods with the CN formulations, similar to Chapter 4. For example, for the CNAD method of Eqn. (5-11), the perturbation error in Eqn. (5-11a) is $b^2 D_x D_y (E_x^{n+1} - E_x^n)$, which is 2nd order accurate. The CNCS-EIX method is formulated by adding two terms, $(c\Delta t/2)^4 D_{3x} D_y (E_y^{n+1} - E_y^n)$ and $(c\Delta t/2)^4 D_{2x} D_{2y} (E_x^{n+1} - E_x^n)$ to Eqn. (5-25b), which are 4th order accurate in time; D_{3x} is the central difference operator for the 3rd derivative along x-axis. The perturbation errors for other CNCS methods can be found similarly. One can recognize them from the amplification coefficient matrices. Note that the terms higher than the 2nd order do not appear into the update equations. They are implicitly added. This is due to the efficient splitting scheme. All the difference operators are second order accurate in space. Thus all the proposed methods are 2nd order accurate in both time and space.

In terms of numerical dispersion along the diagonals, CNAD (CNDS and ADI) is more accurate than CN and CNCS (CNDG). In terms of anisotropy, CN and CNCS (CNDG) are more accurate than CNAD (CNDS and ADI). From the perturbation errors, it can be seen that ADI, CNAD and CNDS are formulated by adding 2nd- and higher-order terms. However, CNCS and CNDG are formulated by adding 4th order terms, which slightly modifies the CN scheme, whereas CN has larger numerical dispersion error along the diagonals than ADI. Thus CNCS and CNDG preserve this property of CN. To reduce the numerical dispersion error, some 2nd order terms can be added to further modify the CN-based methods.

The large numerical dispersion at large Courant number for CNDG and CNCS is originated from the decoupling by adding higher-order terms. It may be possible to realize the decoupling using block factorization [145], thus preserve the numerical dispersion relation of the original CN scheme, at a cost of longer CPU time.

5.8.4 Count of floating-point operations

Both the CNAD and the CNCS (CNDG) methods solve two tri-diagonal matrices at each time step, as does the ADI-FDTD method. However, the number of floating-point arithmetic operations in the right hand side of the update equations is different. Since the multiplication/division (M/D) operation takes more CPU time than addition/subtraction (A/S), they are counted separately. Table 5-1 shows the count of arithmetic operations for ADI, CNAD, CNDG, CNCS-EI and CNCS-IE for one update cycle. In counting operations, the update equations have been rearranged to minimize the number of multiplications for CNCS-EI. For example, both sides of an equation can often be divided by a common coefficient to reduce the number of multiplications required. Note the actual computation efficiency may be code-dependent, and also dependent on the optimization performed by the compiler. Table 5-1 shows that CNDG uses the fewest arithmetic operations, CNAD and CNCS use fewer multiplications than ADI, but more additions. However, ADI uses one more for-loop than CNDG and CNCS, and two more for-loops than CNAD. The overhead of additional for-loops consume more CPU time, particularly for large problems.

Table 5-1 Count of arithmetic operations for different algorithms.

Method	Implicit		Explicit		Total		For-loops
	M/D	A/S	M/D	A/S	M/D	A/S	
ADI	3+3	6+6	1+2+1+2	2+4+2+4	12	24	6
CNDG	6	12	1+1	4+4	8	20	4+1*
CNAD CNDS	3+3	8+12	2	8	8	28	3+1*
CNCS_EI	1+2	3+11	3+3	8+12	9	34	4+1*
CNCS_IE	3+3	8+12	2+1	8+8	9	36	4+1*

* One for-loop is required to store the values at time step n .

5.9 Summary

This Chapter has introduced several CN-based unconditionally-stable methods for efficiently solving Maxwell's Equations for a 2D TE_z wave. All the algorithms have been numerically tested by coding them. The numerical results agree well with the theoretical prediction for the numerical dispersion relation. The CNCS and CNDG methods have non-physical intermediate field values, whereas the CNAD and CNDS methods do not have. However, in the CNCS and CNDG methods, those field values are parts of the methods, and are integrated in the scheme. Unlike ADI-FDTD, these intermediate values will not propagate into the enclosing mesh in the sub-griding method.

In addition, the magnitudes of the amplification factors are unity, thus all the methods presented in this chapter are unconditionally stable. In terms of floating-point arithmetic operations, CNDG uses the least CPU time compared to ADI, CNAD and CNCS. CNAD and CNCS uses fewer multiplication operations but more addition operations than ADI, and have one or two for-loops fewer than ADI. The perturbation errors to the CN scheme are the factorization errors, which are 4^{th} order in time for CNCS and CNDG, and 2^{nd} order for CNDS and CNAD. The numerical dispersion is the same along the axes for ADI, CNAD, CNDS, CNDG and CNCS. However, the numerical anisotropy of CNDG

and CNCS is two orders of magnitude smaller than ADI and CNAD (CNDS). Most importantly, there is a specific time step size for a given mesh that the numerical velocity is isotropic, which has not been found in other methods reported previously. Such properties can be best used to remove the numerical dispersion using the coefficient-modification technique. Note that the CNAD methods are the same as [150], but Reference [150] does not give the expression of the amplification factor and the numerical dispersion relation.

Note that the methods of solving for the magnetic field component in the TE_z wave case can be used in solving for the electric field component in the TM_z wave case, where PEC boundaries are easier to treat.

Since the 3D cases are most often encountered in practice, next chapter will introduce some CN-based unconditionally-stable methods for 3D meshes.

Chapter 6 3D Crank-Nicolson-Based Methods

In the 3D case, all the six electromagnetic field components are involved in the discretized Maxwell's Equations, which makes the solution more complicated. This chapter proposes some methods to efficiently implement the CN schemes. Similar to the 2D case, some methods are similar to ADI-FDTD; others are more like the CN itself in terms of anisotropy.

6.1 3D Crank-Nicolson Formulation for FDTD

The 3D CN formulation for Maxwell's Equations can be written as

$$E_x^{n+1} = E_x^n + a_1\left(D_y H_z^{n+1} - D_z H_y^{n+1} + D_y H_z^n - D_z H_y^n\right) \tag{6-1a}$$

$$E_y^{n+1} = E_y^n + a_1\left(D_z H_x^{n+1} - D_x H_z^{n+1} + D_z H_x^n - D_x H_z^n\right) \tag{6-1b}$$

$$E_z^{n+1} = E_z^n + a_1\left(D_x H_y^{n+1} - D_y H_x^{n+1} + D_x H_y^n - D_y H_x^n\right) \tag{6-1c}$$

$$H_x^{n+1} = H_x^n + a_2\left(D_z E_y^{n+1} - D_y E_z^{n+1} + D_z E_y^n - D_y E_z^n\right) \tag{6-1d}$$

$$H_y^{n+1} = H_y^n + a_2\left(D_x E_z^{n+1} - D_z E_x^{n+1} + D_x E_z^n - D_z E_x^n\right) \tag{6-1e}$$

$$H_z^{n+1} = H_z^n + a_2\left(D_y E_x^{n+1} - D_x E_y^{n+1} + D_y E_x^n - D_x E_y^n\right) \tag{6-1f}$$

where the locations of the field components are omitted for brevity, and are the same as used in the Yee's FDTD [14]: $E_x(i+1/2,j,k)$, $E_y(i,j+1/2,k)$, $E_z(i,j,k+1/2)$, $H_x(i,j+1/2,k+1/2)$, $H_y(i+1/2,j,k+1/2)$, and $H_z(i+1/2,j+1/2,k)$. The amplification coefficient matrix for 3D CN can be written using the Fourier method as

$$Q_{CN} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \tag{6-2a}$$

where

$$Q_1 = Q_2 = \begin{pmatrix} 1-\xi & 0 & 0 \\ 0 & 1-\xi & 0 \\ 0 & 0 & 1-\xi \end{pmatrix} \tag{6-2c}$$

$$Q_{12} = a_2(\xi+1)Q_{eh}$$

$$Q_{21} = a_2(\xi+1)Q_{eh}^T$$

$$Q_{eh}=\begin{pmatrix} 0 & J2\sin(\beta_z\Delta z/2)/\Delta z & -J2\sin(\beta_y\Delta y/2)/\Delta y \\ -J2\sin(\beta_z\Delta z/2)/\Delta z & 0 & J2\sin(\beta_x\Delta x/2)/\Delta x \\ J2\sin(\beta_y\Delta y/2)/\Delta y & -J2\sin(\beta_x\Delta x/2)/\Delta x & 0 \end{pmatrix} \tag{6-2d}$$

The amplification polynomial can be obtained from the determinant of Q_{CN}, given by

$$P(\xi)=(1-\xi)^2\left((1+r_x^2+r_y^2+r_z^2)\xi+(1-r_x^2-r_y^2-r_z^2)+4\sqrt{r_x^2+r_y^2+r_z^2}\right)^2 \left((1+r_x^2+r_y^2+r_z^2)\xi+(1-r_x^2-r_y^2-r_z^2)-4\sqrt{r_x^2+r_y^2+r_z^2}\right)^2 \tag{6-2e}$$

where $r_z=c\Delta t\sin(\beta_z\Delta z/2)/\Delta z$, and $\beta_x=\cos(\theta)\cos(\varphi)$, $\beta_y=\cos(\theta)\sin(\varphi)$, and $\beta_z=\sin(\theta)$. The non-stationary amplification factor is

$$\xi=e^{\pm J2\tan^{-1}\left(\sqrt{r_x^2+r_y^2+r_z^2}\right)} \tag{6-3}$$

Thus the numerical dispersion relation can be derived as

$$\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}=\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2} \tag{6-4}$$

where $\beta=\sqrt{\beta_x^2+\beta_y^2+\beta_z^2}$ is the numerical wave phase constant. It can be seen that 3D CN has a unity-magnitude amplification factor, thus is unconditionally stable. However, the formulation Eqn. (6-1) requires solving a huge, sparse, banded matrix at each time step. For many practical problems, the CPU time consumed to directly solve such a matrix is so long that its advantage of using large time step size is lost. This chapter presents methods to solve Eqn. (6-1) efficiently with some modifications. For convenience, Eqn. (6-1) is written in matrix form using the difference operator matrices as

$$(I-D_1-D_2)W^{n+1}=(I+D_1+D_2)W^n \tag{6-5}$$

where
$$D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_1 D_z & 0 \\ 0 & 0 & 0 & 0 & 0 & -a_1 D_x \\ 0 & 0 & 0 & -a_1 D_y & 0 & 0 \\ 0 & 0 & -a_2 D_y & 0 & 0 & 0 \\ -a_2 D_z & 0 & 0 & 0 & 0 & 0 \\ 0 & -a_2 D_x & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-6a}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & a_1 D_y \\ 0 & 0 & 0 & a_1 D_z & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 D_x & 0 \\ 0 & a_2 D_z & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 D_x & 0 & 0 & 0 \\ a_2 D_y & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-6b}$$

Similar to the 2D case in Chapter 5, the factorized CN scheme can be written as

$$(I - D_1)(I - D_2)W^{n+1} = (I + D_1)(I + D_2)W^n \tag{6-7}$$

The factorization error is $D_1 D_2 W^{n+1} - D_1 D_2 W^n$. However, different from the 2D case, analysis shows that every equation in Eqn. (6-1) has factorization error, which is given by

$$D_1 D_2 = \begin{pmatrix} \Delta D & 0 \\ 0 & \Delta D^T \end{pmatrix} \tag{6-8a}$$

$$\Delta D = \begin{pmatrix} 0 & 0 & -b^2 D_x D_z \\ -b^2 D_x D_y & 0 & 0 \\ 0 & -b^2 D_y D_z & 0 \end{pmatrix} \tag{6-8b}$$

The construction of Eqn. (6-6) is important, because different forms of matrices lead to quite different solutions and stability conditions. For example, the matrices can be constructed by LU decomposition by writing

$$D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_1 D_z & a_1 D_y \\ 0 & 0 & 0 & a_1 D_z & 0 & -a_1 D_x \\ 0 & 0 & 0 & -a_1 D_y & a_1 D_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-9a}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_2 D_z & -a_2 D_y & 0 & 0 & 0 \\ -a_2 D_z & 0 & a_2 D_x & 0 & 0 & 0 \\ a_2 D_y & -a_2 D_x & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-9b}$$

an explicit method is obtained. It is conditionally stable, and has the same Courant limit as the 3D Yee's FDTD.

6.2 A Direct-Splitting Method

Using the direct-splitting method Eqn. (5-1) and difference matrices Eqn. (6-6), the following equations of the direct splitting method of Eqn. (6-7) can be obtained

$$E_x^* + a_1 D_z H_y^* = E_x^n + a_1 \left(2D_y H_z^n - D_z H_y^n\right) \tag{6-10a}$$
$$E_y^* + a_1 D_x H_z^* = E_y^n + a_1 \left(2D_z H_x^n - D_x H_z^n\right) \tag{6-10b}$$
$$E_z^* + a_1 D_y H_x^* = E_z^n + a_1 \left(2D_x H_y^n - D_y H_x^n\right) \tag{6-10c}$$
$$H_x^* + a_2 D_y E_z^* = H_x^n + a_2 \left(2D_z E_y^n - D_y E_z^n\right) \tag{6-10d}$$
$$H_y^* + a_2 D_z E_x^* = H_y^n + a_2 \left(2D_x E_z^n - D_z E_x^n\right) \tag{6-10e}$$
$$H_z^* + a_2 D_x E_y^* = H_z^n + a_2 \left(2D_y E_x^n - D_x E_y^n\right) \tag{6-10f}$$

$$E_x^{n+1} - a_1 D_y H_z^{n+1} = E_x^* - a_1 D_y H_z^n \tag{6-11a}$$
$$E_y^{n+1} - a_1 D_z H_x^{n+1} = E_y^* - a_1 D_z H_x^n \tag{6-11b}$$
$$E_z^{n+1} - a_1 D_x H_y^{n+1} = E_z^* - a_1 D_x H_y^n \tag{6-11c}$$
$$H_x^{n+1} - a_2 D_z E_y^{n+1} = H_x^* - a_2 D_z E_y^n \tag{6-11d}$$
$$H_y^{n+1} - a_2 D_x E_z^{n+1} = H_y^* - a_2 D_x E_z^n \tag{6-11e}$$
$$H_z^{n+1} - a_2 D_y E_x^{n+1} = H_z^* - a_2 D_y E_x^n \tag{6-11f}$$

Inserting Eqn. (6-10e) to Eqn. (6-10a), Eqn. (6-10f) to Eqn.(6-10b), and Eqn. (6-10d) to Eqn. (6-10c) obtains the following intermediate update equations at the intermediate time step

$$(1-b^2D_{2z})E_x^* = E_x^n + 2a_1\left(D_yH_z^n - D_zH_y^n\right) - b^2\left(2D_xD_zE_z^n - D_{2z}E_x^n\right) \tag{6-12a}$$

$$(1-b^2D_{2x})E_y^* = E_y^n + 2a_1\left(D_zH_x^n - D_xH_z^n\right) - b^2\left(2D_xD_yE_x^n - D_{2x}E_y^n\right) \tag{6-12b}$$

$$(1-b^2D_{2y})E_z^* = E_z^n + 2a_1\left(D_xH_y^n - D_yH_x^n\right) - b^2\left(2D_yD_zE_y^n - D_{2y}E_z^n\right) \tag{6-12c}$$

Inserting Eqn. (6-11f) into Eqn. (6-11a), Eqn. (6-11f) to Eqn.(6-11b), and Eqn. (6-11d) to Eqn. (6-11c) obtains the following update equations using the intermediate values

$$(1-b^2D_{2y})E_x^{n+1} = E_x^* + a_1D_y(H_z^* - H_z^n) - b^2D_{2y}E_x^n \tag{6-13a}$$

$$(1-b^2D_{2z})E_y^{n+1} = E_y^* + a_1D_z(H_x^* - H_x^n) - b^2D_{2z}E_y^n \tag{6-13b}$$

$$(1-b^2D_{2x})E_z^{n+1} = E_z^* + a_1D_x(H_y^* - H_y^n) - b^2D_{2x}E_z^n \tag{6-13c}$$

This method is termed as 3D "direct-splitting" method for the CN formulation (CNDS). It can be seen that at each time step, in total six tridiagonal matrices need to be solved for the electric field components, and six explicit equations are required for the magnetic field components. Since the magnetic field components at the intermediate time step in Eqs. (6-10d), (6-10e) and (6-10f) are explicit, they can be substituted into Eqs. (6-11d), (6-11e) and (6-11f) directly, which leads to

$$(1-b^2D_{2y})E_x^{n+1} = E_x^* + b^2D_{2y}E_x^n - b^2D_xD_y(E_y^* + E_y^n) \tag{6-14a}$$

$$(1-b^2D_{2z})E_y^{n+1} = E_y^* + b^2D_{2z}E_y^n - b^2D_yD_z(E_z^* + E_z^n) \tag{6-14b}$$

$$(1-b^2D_{2x})E_z^{n+1} = E_z^* + b^2D_{2x}E_z^n - b^2D_xD_z(E_x^* + E_x^n) \tag{6-14c}$$

Thus the six explicit update equations can be reduced to three as follows

$$H_x^{n+1} = H_x^n + a_2\left\{D_z(E_y^{n+1} + E_y^n) - D_y(E_z^* + E_z^n)\right\} \tag{6-15a}$$

$$H_y^{n+1} = H_y^n + a_2\left\{D_x(E_z^{n+1} + E_z^n) - D_z(E_x^* + E_x^n)\right\} \tag{6-15b}$$

$$H_z^{n+1} = H_z^n + a_2\left\{D_y(E_x^{n+1} + E_x^n) - D_x(E_y^* + E_y^n)\right\} \tag{6-15c}$$

Therefore the complete CNDS algorithm can be performed in the order of Eqs. (6-12), (6-14) implicitly, and (6-15) explicitly. This formulation not only reduces the count of arithmetic operations, but also eliminates three for-loops.

Note that from Eqs. (6-10) and (6-11), the magnetic field components can be also solved implicitly if necessary.

With the Fourier analysis method and some complicated manipulation, the amplification polynomial can be obtained as follows

$$P(\xi) = (1-\xi)^2 \xi^2 \left(p_2 (\xi^2 + \xi^{-2}) + p_1 (\xi + \xi^{-1}) + p_0 \right) \tag{6-16}$$

where p_0, p_1 and p_2 are polynomials of r_x^2, r_y^2 and r_z^2, which will not be given here due to complexity. With some manipulation, the amplification factor of the non-stationary mode can be found explicitly as

$$\xi = \exp\left(\pm J \tan^{-1} \phi \right) \tag{6-17a}$$

$$\phi = \frac{\sqrt{(1+r_x^2)^2 (1+r_y^2)^2 (1+r_z^2)^2 - (1-r_x^2 - r_y^2 - r_z^2 - r_x^2 r_y^2 - r_x^2 r_z^2 - r_y^2 r_z^2 + r_x^2 r_y^2 r_z^2)^2}}{(1-r_x^2 - r_y^2 - r_z^2 - r_x^2 r_y^2 - r_x^2 r_z^2 - r_y^2 r_z^2 + r_x^2 r_y^2 r_z^2)^2} \tag{6-17b}$$

The numerical dispersion relation thus can be derived as

$$\begin{aligned} &\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2} \left(1 + (c\Delta t)^6 \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2} \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} \frac{\sin^2(\beta_z \Delta z/2)}{\Delta z^2} \right) \\ &= \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2} + \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} + \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} + (c\Delta t)^2 \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2} \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} \\ &+ (c\Delta t)^2 \frac{\sin^2(\beta_x \Delta x/2)}{\Delta x^2} \frac{\sin^2(\beta_z \Delta z/2)}{\Delta z^2} + (c\Delta t)^2 \frac{\sin^2(\beta_y \Delta y/2)}{\Delta y^2} \frac{\sin^2(\beta_z \Delta z/2)}{\Delta z^2} \end{aligned} \tag{6-18}$$

Eqn. (6-18) is recognized as the same as the numerical dispersion relation Eqn. (4-29) of 3D ADI-FDTD. However, the count of floating-point-arithmetic operations and thus the CPU time are different from ADI-FDTD as demonstrated in Section 6.7.3 below.

6.3 Coupled Equation for the Unknown Electric Field Components

Since the CNDS method has larger numerical anisotropy than the original CN formulation, some other more isotropic algorithms must be sought. From Chapter 5 it is known that if the unknown magnetic field components at time step n+1 H^{n+1} are eliminated, then several efficient solution methods can be devised in solving for the coupled electric field components. With some complicated manipulation the coupled electric field components can be written

$$\begin{aligned}&\left(1-b^2(D_{2y}+D_{2z})\right)E_x^{n+1}+b^2D_xD_yE_y^{n+1}+b^2D_xD_zE_z^{n+1} \\ &=E_x^n+2a_1\left(D_yH_z^n-D_zH_y^n\right)+b^2\left(D_{2y}+D_{2z}\right)E_x^n-b^2D_xD_yE_y^n-b^2D_xD_zE_z^n\end{aligned} \tag{6-19a}$$

$$\begin{aligned}&\left(1-b^2(D_{2x}+D_{2z})\right)E_y^{n+1}+b^2D_xD_yE_x^{n+1}+b^2D_yD_zE_z^{n+1} \\ &=E_y^n+2a_1\left(D_zH_x^n-D_xH_z^n\right)+b^2\left(D_{2x}+D_{2z}\right)E_y^n-b^2D_xD_yE_x^n-b^2D_yD_zE_z^n\end{aligned} \tag{6-19b}$$

$$\begin{aligned}&\left(1-b^2(D_{2x}+D_{2y})\right)E_z^{n+1}+b^2D_xD_zE_x^{n+1}+b^2D_yD_zE_y^{n+1} \\ &=E_z^n+2a_1\left(D_xH_y^n-D_yH_x^n\right)+b^2\left(D_{2x}+D_{2y}\right)E_z^n-b^2D_xD_zE_x^n-b^2D_yD_zE_y^n\end{aligned} \tag{6-19c}$$

Eqn. (6-19) can be written in a concise matrix as

$$(I-A-B)W^{n+1}=(I+A+B)W^n+f_e \tag{6-20a}$$

where

$$f_e=2a_1\begin{pmatrix}0 & -D_z & D_y\\ D_z & 0 & -D_x\\ -D_y & D_x & 0\end{pmatrix}\begin{pmatrix}H_x\\ H_y\\ H_z\end{pmatrix} \tag{6-20b}$$

Similar to the 2D case, Eqn. (6-20a) can be factorized and split as

$$(I-A)W^*=(I+A+2B)W^n+f_e \tag{6-21a}$$

$$(I-B)W^{n+1}=W^*-BW^n \tag{6-21b}$$

Depending on the formulation of the difference matrices A and B, different algorithms can be obtained. Section 6.4 introduces several cycle-sweep methods [163] and Section 6.5 proposes approximate-factorization-splitting method [167].

6.4 Cycle-Sweep Methods in 3D

Now the matrices A and B are 3 by 3, therefore there are more degrees of freedom in the formulation. This Section gives two of such formulations.

6.4.1 Cycle-sweep-uniform method

By constructing the following matrices

$$A = b^2 \begin{pmatrix} D_{2y} & 0 & 0 \\ -D_x D_y & D_{2z} & 0 \\ -D_x D_z & -D_y D_z & D_{2x} \end{pmatrix} \tag{6-22a}$$

$$B = b^2 \begin{pmatrix} D_{2z} & -D_x D_y & -D_x D_z \\ 0 & D_{2x} & -D_y D_z \\ 0 & 0 & D_{2y} \end{pmatrix} \tag{6-22b}$$

Eqn. (6-21) can be solved in a cycle-sweep manner [163], according to Eqn. (6-21), as

$$(1-b^2 D_{2y})E_x^* = (1+b^2 D_{2y}+2b^2 D_{2z})E_x^n - 2b^2 D_x D_y E_y^n - 2b^2 D_x D_z E_z^n + f_{e1} \tag{6-23a}$$

$$(1-b^2 D_{2z})E_y^* = (1+b^2 D_z^2+2b^2 D_{2x})E_y^n - b^2 D_x D_y (E_x^* + E_x^n) - 2b^2 D_y D_z E_z^n + f_{e2} \tag{6-23b}$$

$$(1-b^2 D_{2x})E_z^* = (1+b^2 D_{2x}+2b^2 D_{2y})E_z^n - b^2 D_x D_z (E_x^* + E_x^n) - b^2 D_y D_z (E_y^* + E_x^n) + f_{e3} \tag{6-23c}$$

$$(1-b^2 D_{2y})E_z^{n+1} = E_z^* - b^2 D_{2y} E_z^n \tag{6-23d}$$

$$(1-b^2 D_{2x})E_y^{n+1} = E_y^* - b^2 D_{2x} E_y^n + b^2 D_y D_z (E_z^n - E_z^{n+1}) \tag{6-23e}$$

$$(1-b^2 D_{2z})E_x^{n+1} = E_x^* - b^2 D_{2z} E_x^n + b^2 D_x D_y (E_y^n - E_y^{n+1}) + a_1 a_2 D_x D_z (E_z^n - E_z^{n+1}) \tag{6-23f}$$

It can be seen that Eqn. (6-23) solves six tridiagonal matrices at each time step. The magnetic field components can be found explicitly using Eqs. (6-1d), (6-1e) and (6-1f). The update cycle for Eqn. (6-23) starts with E_x, sweeps through E_y and E_z, and ends at E_x, and "uniformly" solves a tridiagonal matrix for each equation corresponding to each electrical field component, thus is termed as the "Crank-Nicolson-cycle-sweep-uniform" (CNCSU) method. It can also start and end at E_y or E_z if desired. Note that there are intermediate values E_x^*, E_y^* and E_z^* which are non-physical, as in ADI-FDTD, except

that they are not associated with a specific time value. The perturbation errors are the factorization error AB.

In addition, there are several different formulations of A and B, leading to slightly different update equations. Next sub-section introduces some other cycle-sweep methods similar to those in 2D case.

6.4.2 Implicit/explicit and explicit/implicit cycle-sweep method

The construction of Eqn. (6-22) allows the implicit 2nd-degree derivatives to be uniformly distributed into A and B. Similar to the 2D case, one can also construct A and B such that all the 2nd order derivatives have been collected into one matrix and the cross-derivatives uniformly distributed in both matrices. For example, beginning with and ending at E_x, A and B can be constructed as

$$A = b^2 \begin{pmatrix} D_{2y} + D_{2z} & 0 & 0 \\ -D_x D_y & D_{2y} + D_{2z} & 0 \\ -D_x D_z & -D_y D_z & D_{2x} + D_{2y} \end{pmatrix} \tag{6-24a}$$

$$B = b^2 \begin{pmatrix} 0 & -D_x D_y & -D_x D_z \\ 0 & 0 & -D_y D_z \\ 0 & 0 & 0 \end{pmatrix} \tag{6-24b}$$

However, to solve for the intermediate values $(I - A)W^*$, the two 2nd-degree derivatives must be factorized for an efficient solution by adding the following matrix

$$\Delta A = b^4 \begin{pmatrix} D_{2y} D_{2z} & 0 & 0 \\ 0 & D_{2y} D_{2z} & 0 \\ 0 & 0 & D_{2x} D_{2y} \end{pmatrix} \tag{6-25}$$

This method is termed as the "Crank-Niclolson-cycle-sweep-implicit-explict-Ex" (CNCSIE-X) method, and is formulated as

$$(I-A+\Delta A)W^{*}=(I+A+2B+\Delta A)W^{n}+f_e \tag{6-26a}$$

$$(I-B)W^{n+1}=W^{*}-BW^{n} \tag{6-26b}$$

where f_e is given by Eqn. (6-12b). The update equations are

$$\left(1-b^2D_{2y}\right)E_x^{*}=(1+b^2D_{2y}+2b^2D_{2z})E_x^{n}-2b^2D_xD_yE_y^{n}-2b^2D_xD_zE_z^{n}+f_{e1} \tag{6-27a}$$

$$\left(1-b^2D_{2z}\right)E_x^{**}=E_x^{*}-b^2D_{2z}E_x^{n} \tag{6-27b}$$

$$\left(1-b^2D_{2z}\right)E_y^{*}=(1+b^2D_z^2+2b^2D_{2x})E_y^{n}-b^2D_xD_y(E_x^{**}+E_x^{n})-2b^2D_yD_zE_z^{n}+f_{e2} \tag{6-27c}$$

$$\left(1-b^2D_{2x}\right)E_y^{**}=E_y^{*}-b^2D_{2x}E_y^{n} \tag{6-27d}$$

$$\left(1-b^2D_{2x}\right)E_z^{*}=(1+b^2D_{2x}+2b^2D_{2y})E_z^{n}-b^2D_xD_z(E_x^{**}+E_x^{n})-b^2D_yD_z(E_y^{**}+E_x^{n})+f_{e3} \tag{6-27e}$$

$$\left(1-b^2D_{2y}\right)E_z^{n+1}=E_z^{*}-b^2D_{2y}E_z^{n} \tag{6-27f}$$

$$E_y^{n+1}=E_y^{**}+D_yD_z(E_z^{n}-E_z^{n+1}) \tag{6-27g}$$

$$E_x^{n+1}=E_x^{**}+D_xD_y(E_y^{n}-E_y^{n+1})+D_xD_z(E_z^{n}-E_z^{n+1}) \tag{6-27h}$$

Similarly, one can construct the Crank-Niclolson-cycle-sweep-explict-implicit (CNCSEI) method by changing the elements in matrices A and B. Note that Eqn. (6-26a) itself can be used to approximately solve Eqn. (6-19), which is the 3D "approximate-decoupling-Douglas-Gunn" (CNADDG) method similar to CNAD in the 2D case. Since CNADDG sweeps electric field components one by one, thus uses two intermediate arrays fewer than CNCSU methods. The overall perturbation errors for CNCSIE are

$$\Delta D_LW^{n+1}-\Delta D_RW^{n} \tag{6-28a}$$

$$\Delta D_L=\begin{pmatrix}\Delta D_{L11} & O\\ O & O\end{pmatrix}\qquad \Delta D_R=\begin{pmatrix}\Delta D_{R11} & O\\ O & O\end{pmatrix} \tag{6-28b}$$

$$\Delta D_{L11}=\begin{pmatrix}-D_{2y}D_{2z} & D_xD_{3y}+D_xD_yD_{2z}-D_xD_{3y}D_{2z} & D_xD_{2y}D_z+D_xD_{3z}-D_xD_{2y}D_{3z}\\ 0 & -D_{2x}D_{2y}-D_{2x}D_{2z} & D_yD_{3z}-D_{2x}D_yD_{3z}\\ 0 & -D_{2x}D_yD_z & -D_{2x}D_{2y}-D_{2x}D_{2z}-D_{2y}D_{2z}\end{pmatrix} \tag{6-28c}$$

$$\Delta D_R=\begin{pmatrix}-D_{2y}D_{2z} & D_xD_{3y}+D_xD_yD_{2z}+D_xD_{3y}D_{2z} & D_xD_{2y}D_z+D_xD_{3z}+D_xD_{2y}D_{3z}\\ 0 & -D_{2x}D_{2y}-D_{2x}D_{2z} & D_yD_{3z}+D_{2x}D_yD_{3z}\\ 0 & -D_{2x}D_yD_z & -D_{2x}D_{2y}-D_{2x}D_{2z}-D_{2y}D_{2z}\end{pmatrix} \tag{6-28d}$$

It can be seen that the lowest errors are 4th order, similar to CNCSU method.

6.4.3 Amplification polynomial and amplification factor for CNCSU

For the CNCSU method, the implicit update equations Eqn. (6-23) have been added some higher-order terms from the factorization, and those higher-order terms are avoided from the splitting in order to minimizing the computational stencil. Using the Fourier method, the following amplification coefficient matrix can be built up

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \tag{6-29a}$$

where

$$Q_1 = \begin{pmatrix} (1-r_y^2)(1-r_z^2)-(1+r_y^2)(1+r_z^2)\xi & r_x r_y(1-r_y^2)+r_x r_y(1+r_y^2)\xi & r_x r_z(1-r_y^2)+r_x r_z(1+r_y^2)\xi \\ r_x r_y(1-r_z^2)+r_x r_y(1+r_z^2)\xi & (1-r_x^2)(1-r_z^2)-(1+r_x^2)(1+r_z^2)\xi+r_x^2 r_y^2(1-\xi) & r_y r_z(1+r_x^2-r_z^2)+r_y r_z(1-r_x^2+r_z^2)\xi \\ r_x r_z(1-r_z^2)+r_x r_z(1+r_z^2)\xi & r_y r_z(\xi+1) & (1-r_x^2)(1-r_y^2)+(r_x^2 r_z^2+r_y^2 r_z^2)(1-\xi)-(1+r_x^2)(1+r_y^2)\xi \end{pmatrix} \tag{6-29b}$$

$$Q_{eh} = \begin{pmatrix} 0 & J2\sin(\beta_z \Delta z/2)/\Delta z & -J2\sin(\beta_y \Delta y/2)/\Delta y \\ -J2\sin(\beta_z \Delta z/2)/\Delta z & 0 & J2\sin(\beta_x \Delta x/2)/\Delta x \\ J2\sin(\beta_y \Delta y/2)/\Delta y & -J2\sin(\beta_x \Delta x/2)/\Delta x & 0 \end{pmatrix} \tag{6-29c}$$

$$Q_{12} = 2a_1 Q_{eh} \tag{6-29d}$$

$$Q_{21} = a_2(\xi+1)Q_{eh}^T \tag{6-29e}$$

$$Q_{22} = (1-\xi)I \tag{6-19f}$$

With the aid of the symbolic algebra program Maple V, after complicated manipulation and re-arrangement, the amplification polynomial can be written as

$$P(\xi) = (1-\xi)^2 \xi^2 \left(p_2(\xi^2 + \xi^{-2}) + p_1(\xi + \xi^{-1}) + p_0 \right) \tag{6-30a}$$

$$\begin{aligned} p_2 = {} & (1+r_x^4)(1+r_y^4)(1+r_z^4) + 2(r_x^2 + r_y^2 + r_z^2) + 4(r_x^2 r_y^2 + r_x^2 r_z^2 + r_y^2 r_z^2) + 8 r_x^2 r_y^2 r_z^2 \\ & + 2(r_x^2(r_y^4 + r_z^4) + r_y^2(r_x^4 + r_z^4) + r_z^2(r_x^4 + r_y^4)) \\ & + 4(r_x^2 r_y^2 r_z^4 + r_x^2 r_y^4 r_z^2 + r_x^4 r_y^2 r_z^2) + 2(r_x^2 r_y^4 r_z^4 + r_x^4 r_y^2 r_z^4 + r_x^4 r_y^4 r_z^2) \end{aligned} \tag{6-30b}$$

$$p_1 = -4(1-r_x^4)(1-r_y^4)(1-r_z^4) - 8 r_x^4 r_y^4 r_z^4 \tag{6-30c}$$

$$\begin{aligned} p_0 = {} & 6(1+r_x^4)(1+r_y^4)(1+r_z^4) - 4r_x^2(1+r_y^4+r_z^4+r_y^4 r_z^4) - 4r_y^2(1+r_x^4+r_z^4+r_x^4 r_z^4) \\ & - 4r_z^2(1+r_x^4+r_y^4+r_x^4 r_y^4) - 16 r_x^2 r_y^2 r_z^2 + 24(r_x^2 r_y^2(1+r_z^4) + r_x^2 r_z^2(1+r_y^4) + r_y^2 r_z^2(1+r_x^4)) \end{aligned} \tag{6-30d}$$

After eliminating the stationary solutions, the following expression can be obtained

$$\xi + \xi^{-1} = \frac{-p_1 \pm \sqrt{p_1^2 - 4p_0 p_2}}{2p_2} \tag{6-31a}$$

$$p_1^2 - 4p_0 p_2 = 64 r_x^4 r_y^4 r_z^4 \left((1-r_x^4)(1-r_y^4)(1-r_z^4) + r_x^4 r_y^4 r_z^4\right) \tag{6-31b}$$

It can be shown that Eqn. (6-31b) is non-negative, provided that the following conditions are satisfied

$$r_x^4 \le 1 \qquad r_y^4 \le 1 \qquad r_z^4 \le 1 \tag{6-32}$$

Suppose the smallest spatial size is $\Delta_{\min} = \min(\Delta x, \Delta y, \Delta z)$, then Eqn. (6-32) is simplified to be $r_{\Delta_{\min}}^2 \le 1$. Thus $\frac{s_{\min}}{2}\sin(\frac{\beta\Delta_{\min}}{2}) \le \frac{s_{\min}}{2}\frac{\beta\Delta_{\min}}{2} = \frac{\omega\Delta t}{4} \le 1$, where $s_{\min} = c\Delta t / \Delta_{\min}$. Recall that the Nyquist criterion can be written as $\omega\Delta t \le \pi$. It can be seen that the conditions Eqn. (6-32) are larger than the Nyquist limit, thus the CNCSU is unconditionally stable. By the use of the Möbius transformation [129], the same condition Eqn. (6-32) can be obtained. In addition, because Eqn. (6-31b) is positive, the CNCSU is also strictly non-dissipative. Since the perturbation errors of CNCSU to the CN scheme are all higher orders, CNCSU has the same unconditional stability as the original CN scheme.

Analysis shows that for a stable scheme, its amplification factor polynomial (excluding the stationary solutions) in 3D should be written in the following form

$$\left(a + b\xi + c\xi^2\right)^2 \tag{6-33}$$

where a, b and c are the coefficients of the polynomial. With this hypothesis, the numerical dispersion relation of the CNCSU can be derived.

Expanding Eqn. (6-33) and comparing to Eqn. (6-30) obtains

$$a^2 = c^2 = p_2 \tag{6-34a}$$

$$2ab = 2bc = p_1 \tag{6-34b}$$

$$2ac + b^2 = p_0 \tag{6-34c}$$

Therefore the amplification factor can be solved as

$$\xi = \frac{-p_1 \pm J\sqrt{16p_2^2 - p_1^2}}{4p_2} \tag{6-35}$$

The numerical dispersion relation for CNCSU can be derived as

$$p_1 = -4p_2 \cos(\omega\Delta t) \tag{6-36}$$

Expanding it obtains

$$\begin{aligned}&\left(1 + r_x^4 r_y^4 + r_x^4 r_x^4 + r_y^4 r_z^4 + r_x^4 r_y^4 r_z^4\right)\sin^2(\omega\Delta t/2) = \left(r_x^4 + r_y^4 + r_z^4\right)\cos^2(\omega\Delta t/2)\\ &+ \{(r_x^2 + r_y^2 + r_z^2) + 2(r_x^2 r_y^2 + r_x^2 r_z^2 + r_y^2 r_z^2) + 4r_x^2 r_y^2 r_z^2 + (r_x^2(r_y^4 + r_z^4) +\\ &r_y^2(r_x^4 + r_z^4) + r_z^2(r_x^2 + r_y^2)) + 2(r_x^2 r_y^2 r_z^4 + r_x^2 r_y^4 r_z^2 + r_x^4 r_y^2 r_z^2)\\ &+ (r_x^2 r_y^4 r_z^4 + r_x^4 r_y^2 r_z^4 + r_x^4 r_y^4 r_z^2)\}\cos(\omega\Delta t)\end{aligned} \tag{6-37}$$

When collapsed to the 2D case, it can be shown that Eqn. (6-37) can be simplified to the relation for 2D CNCS in Eqn. (5-8d).

It can be shown that the expression of amplification factor for CNCSIE is similar to that Eqn. (6-30a) of CNCSU. Thus following the same procedure mentioned above the numerical dispersion relation for CNCSIE can be obtained. Due to its complexity, it will not be given here.

6.5. Approximate-Factorization-Splitting Method

CNCS methods are not suitable for parallel computation because of the alternating sweep. CNDS can be solved in parallel, but the anisotropy is large compared to the original CN formulation. It is possible to devise a method suitable for parallel computation, while maintaining small anisotropy. The construction of the abstract difference operators allows direct algebraic manipulation, taking the operands as common algebraic variables. Therefore, Eqn. (6-19) can be further decomposed. With some complicated manipulation, the update equations for the electric field components can be written [166]

$$\left(1-b^2D_{2x}-b^2D_{2y}-b^2D_{2z}\right)E_x^{n+1}=g_1 \tag{6-38a}$$
$$\left(1-b^2D_{2x}-b^2D_{2y}-b^2D_{2z}\right)E_y^{n+1}=g_2 \tag{6-38b}$$
$$\left(1-b^2D_{2x}-b^2D_{2y}-b^2D_{2z}\right)E_z^{n+1}=g_3 \tag{6-38c}$$

where

$$g_1=E_x^n+2a_1\left(D_yH_z^n-D_zH_y^n\right)+b^2\left(D_{2y}-D_{2x}+D_{2z}\right)E_x^n-2b^2\left(D_xD_yE_y^n+D_xD_zE_z^n\right) \tag{6-39a}$$
$$g_2=E_y^n+2a_1\left(D_zH_x^n-D_xH_z^n\right)+b^2\left(D_{2x}-D_{2y}+D_{2z}\right)E_y^n-2b^2\left(D_xD_yE_x^n+D_yD_zE_z^n\right) \tag{6-39b}$$
$$g_3=E_z^n+2a_1\left(D_xH_y^n-D_yH_x^n\right)+b^2\left(D_{2x}+D_{2y}-D_{2z}\right)E_z^n-2b^2\left(D_xD_zE_x^n+D_yD_zE_y^n\right) \tag{6-39c}$$

Eqn. (6-38) has realized complete decoupling, but each equation is still a large block-tridiagonal matrix with fringes. Modifying (6-38) by adding the following higher order terms $b^4\left(D_{2x}D_{2y}+D_{2x}D_{2z}+D_{2y}D_{2z}-b^2D_{2x}D_{2y}D_{2z}\right)\left(E_i^{n+1}-E_i^n\right)$ where E_i is E_x, E_y or E_z to Eqn. (6-38) allows us to perform approximate factorization to the LHS of Eqn. (6-38). The resulting factorized update equations are

$$\begin{aligned}\left(1-b^2D_{2x}\right)\left(1-b^2D_{2y}\right)\left(1-b^2D_{2z}\right)E_x^{n+1}&=g_1\\&+b^4\left(D_{2x}D_{2y}+D_{2x}D_{2z}+D_{2y}D_{2z}-b^2D_{2x}D_{2y}D_{2z}\right)E_x^n\end{aligned} \tag{6-40a}$$

$$\left(1-b^2D_{2x}\right)\left(1-b^2D_{2y}\right)\left(1-b^2D_{2z}\right)E_y^{n+1} = g_2 + b^4\left(D_{2x}D_{2y}+D_{2x}D_{2z}+D_{2y}D_{2z}-b^2D_{2x}D_{2y}D_{2z}\right)E_y^n \tag{6-40b}$$

$$\left(1-b^2D_{2x}\right)\left(1-b^2D_{2y}\right)\left(1-b^2D_{2z}\right)E_z^{n+1} = g_3 + b^4\left(D_{2x}D_{2y}+D_{2x}D_{2z}+D_{2y}D_{2z}-b^2D_{2x}D_{2y}D_{2z}\right)E_z^n \tag{6-40c}$$

However, such a formulation has $D_{2x}D_{2y}D_{2z}$ terms in the RHS, which involves 27 (known) field components, and thus makes the computational stencil very large. To avoid cross terms in the RHS of Eqn. (6-40), the 3D efficient splitting scheme with some modification can be applied, which leads to

$$(1-b^2D_{2a})E_i^* = g_i \tag{6-41a}$$

$$(1-b^2D_{2b})E_i^{**} = E_i^* - b^2D_{2b}E_i^n \tag{6-41b}$$

$$(1-b^2D_{2c})E_i^{n+1} = E_i^{**} - b^2D_{2c}E_i^n \tag{6-41c}$$

where g_i is g_x, g_y or g_z and is given by

$$g_x = g_1 + b^2\left(D_{2b}+D_{2c}\right)E_x^n \tag{6-42a}$$

$$g_y = g_2 + b^2\left(D_{2b}+D_{2c}\right)E_y^n \tag{6-42b}$$

$$g_z = g_3 + b^2\left(D_{2b}+D_{2c}\right)E_z^n \tag{6-42c}$$

The amplification polynomial is too complicated to write out, but can be solved for the amplification factor. With some manipulation the amplification factor can be concisely expressed as

$$\xi = \frac{R_-^2 \pm J\sqrt{R_+^2 - R_-^2}}{R_+^2} = \exp\left(\pm J\tan^{-1}\frac{\sqrt{R_+^2-R_-^2}}{R_-^2}\right) \tag{6-43}$$

where

$$R_+^2 = (1+r_x^2)^2(1+r_y^2)^2(1+r_z^2)^2 \tag{6-44}$$

$$R_-^2 = \left(1-r_x^2-r_y^2-r_z^2+r_x^2r_y^2+r_x^2r_z^2+r_y^2r_z^2+r_x^2r_y^2r_z^2\right)^2$$

Thus the numerical dispersion relation can be derived as

$$\left(1+(c\Delta t)^6\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\right)\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}$$
$$+(c\Delta t)^4\left(\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\right)\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2}$$
$$+(c\Delta t)^4\left(\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\right)\frac{\tan^2(\omega\Delta t/2)}{(c\Delta t)^2} \tag{6-45}$$
$$=\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}$$

This is expected to be similar to that of 2D CNDG and CNCS. The above formulation requires solving nine tridiagonal matrices. However, by the use of the Divergence Theorem, it can be reduced to six tridiagonal matrices by explicitly solving the third electric field value. Suppose E_x^{n+1} and E_y^{n+1} have been found; then E_z^{n+1} can be found using

$$E_z^{n+1}(i,j,k+1/2)=E_z^{n+1}(i,j,k-1/2)-\left(D_x(E_x^{n+1}+E_x^n)+D_y(E_y^{n+1}+E_y^n)+D_zE_z^n\right) \tag{6-46}$$

This is an upper bi-diagonal matrix, and can be solved very efficiently. In the case of PEC boundary, Eqn. (6-46) can be explicitly solved.

6.6 Implementation of Periodic Boundary Conditions

Sometimes periodic boundary conditions are encountered which have the property that $u(M)=u(1)$. For implicit schemes, the result is a $(M-1)\times(M-1)$ tridiagonal (or block in general) circulant coefficient matrix of the form

$$A=\begin{pmatrix} d & a & 0 & \dots & 0 & b \\ b & d & a & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & b & d & a \\ a & 0 & \dots & 0 & b & d \end{pmatrix} \tag{6-47}$$

The two elements at the up-right and the down-left corners destroy the symmetry of the tridiagonal, thus the usual method in solving the tridiagonal matrix is invalid. To circumvent this difficulty, matrix A can be decomposed into [77]

$$A = B - WZ^T \tag{6-48}$$

where

$$B = \begin{pmatrix} d-a & a & 0 & \dots & 0 & 0 \\ b & d & a & 0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & b & d & a \\ 0 & 0 & \dots & 0 & b & d-b \end{pmatrix} \tag{6-49a}$$

$$W = \begin{pmatrix} 1 & 0 & \dots & 0 & 1 \end{pmatrix}^T \tag{6-49b}$$

$$Z = \begin{pmatrix} a & 0 & \dots & 0 & b \end{pmatrix}^T \tag{6-49c}$$

Then the matrix equation $Ax_0 = b_0$ is transformed into solving the following equations

$$Bx_1 = b_0 \tag{6-50a}$$

$$Bx_2 = W \tag{6-s0b}$$

$$(1 - Z^T x_2) y_0 = Z^T x_1 \tag{6-50c}$$

$$x_0 = x_1 + x_2 y_0 \tag{6-50d}$$

where x_1 and x_2 are (M-1) vectors, and y_0 is scalar quantity. Note that for a given mesh, x_1 and x_2 can be solved for only once at the beginning. Only the scalar y_0 must be computed at each time step. Such a periodic boundary condition is very useful in 2D CNDG and 3D CNAFS, and has been run successfully.

6.7. Discussion

6.7.1 A unified numerical dispersion relation

For comparison, the numerical dispersion relations for CN, CNDS (ADI), and CNAFS can be written in a unified form as

$$\begin{aligned}
&\tan^2(\omega\Delta t/2)+g_1(c\Delta t)^6\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\\
&=(c\Delta t)^2\left(\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\right)\\
&+g_2(c\Delta t)^4\left(\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}+\frac{\sin^2(\beta_x\Delta x/2)}{\Delta x^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\right)\\
&+g_2(c\Delta t)^4\left(\frac{\sin^2(\beta_y\Delta y/2)}{\Delta y^2}\frac{\sin^2(\beta_z\Delta z/2)}{\Delta z^2}\right)
\end{aligned} \tag{6-51}$$

where

$$g_1=\begin{cases}\tan^2(\omega\Delta t/2) & CNDS, ADI\\ 0 & CN\\ \tan^2(\omega\Delta t/2) & CNAFS\end{cases} \tag{6-52a}$$

$$g_2=\begin{cases}1 & CNDS, ADI\\ 0 & CN\\ -\tan^2(\omega\Delta t/2) & CNAFS\end{cases} \tag{6-52b}$$

It can be seen that the terms are the same but the positions of the cross terms are quite different. This makes a huge difference for the anisotropy.

6.7.2 Numerical dispersion and numerical anisotropy for 3D methods

Numerical dispersion along the axes is the same for all methods, which is the same as the 1D ADI method discussed in Chapter 4. However, the numerical dispersion is quite different along the diagonals, thus the anisotropy is different. Fig.6-1 graphs the numerical anisotropy for the 3D versions of CN, CNDS (ADI) and CNAFS at a mesh density from 20 to 50 CPWs, at Courant numbers 1, 2, 3 for a cubic mesh $\Delta x = \Delta y = \Delta z$.

At a mesh density 34 and Courant number 3 (about 5.2 times the Courant limit), the anisotropy of CNAFS is 0.000029, which is near its zero-anisotropy point, whereas ADI has anisotropy of 0.014. The anisotropy of ADI is always larger than that of CN, while for the Courant number below 2, the anisotropy of CNAFS is smaller than that of CN. Note that the anisotropy of CNAFS for Courant number s=3 decreases more rapidly than that of CN. This implies that at some mesh density less than 20 there is a maximum-anisotropy point for CNAFS. This point is near $4.3 \times s = 4.3 \times 3 = 12.9$ CPWs (see Chapter 7, Table 7-3).

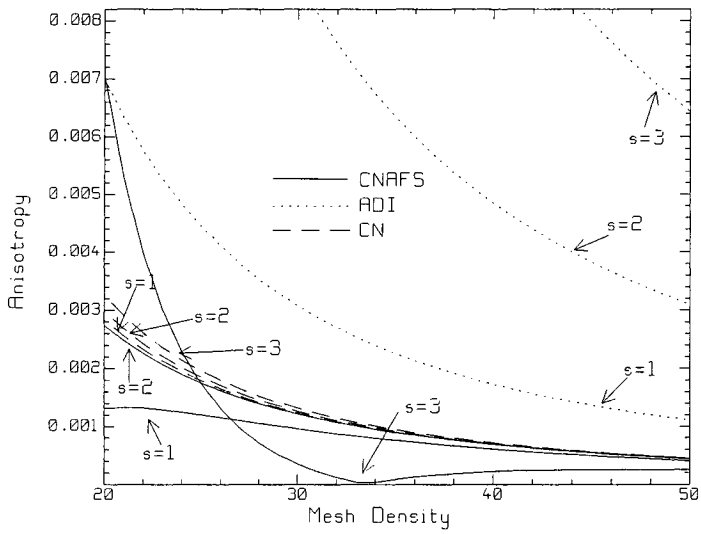

Fig. 6-1 Anisotropy of CN, CNAFS, and ADI for coarse meshes.

Because the CNAFS has much smaller anisotropy, the coefficient-modification introduced in previous chapters is suitable here to remove the numerical dispersion along axes, leading to very small numerical errors in other directions of travel.

6.7.3 Number of floating-point arithmetic operations

Analysis shows that many methods for approximately solving the 3D CN formulation can be devised. However, not only are the numerical errors different, but the

computational efficiency is also different. The number of the floating-point-operations is counted in Table 6-1, in which each field component is counted according to the update equation form. Note that some constants have been divided out of the original update equations to have an optimal count and fast simulation. It can be seen that CNDS and CNCSU have two fewer loops than ADI, thus are usually more efficient than ADI.

Table 6-1 Number of floating-point arithmetic operations

	Implicit		Explicit		Total		Loops
	M/D	A/S	M/D	A/S	M/D	A/S	
ADI	6×3	6×8	6×2	6×4	30	72	12
CNDS	3×(4+3)	3× (10+11)	3×2	3×8	27	87	9+1*
CNCSU	3×7+2+3+4	17+21+25+3+11+19	3×2	3×8	36	110	9+1*
CNADDG	3×(7+2)	17+21+25+3×3	3×2	3×8	33	96	9+1*
CNCSIE	3×(7+2)	17+21+25+3×3	3+3×2	8+16+3×8	36	120	11+1*
CNAFS1	3×(9+4)	3×(20+2×6)	3×2	3×8	45	120	12+1*,**
CNAFS2	2×(9+4)+2	2×(20+2×6)+4	3×2	3×8	34	92	6+1*+1***

* One extra loop to store the current field values into old values.

** Solves nine tridiagonal.

*** Solves six tridiagonal and an extra upper bi-diagonal matrix.

The CPU time consumption for ADI and CNCSU has been tested using a code containing only the for-loops and number of the arithmetic operations. The results show that indeed the two extra for-loops in ADI consume more CPU time than the extra numbers of the arithmetic operations of CNCSU. Taking account of larger CPU time consumptions in solving the tridiagonal matrices, CNCSU, CNDS and CNADDG is as efficient as ADI though they have more counts of the floating-point arithmetic operations.

6.7.4 More concise derivation of the coupled electrical field components

The equations (6-19) for the coupled electric field components can be derived more compactly by writing the Curl equations in CN form as

$$W_e^{n+1} = W_e^n + a_1 \nabla \times \left(W_h^{n+1} + W_h^n \right) \tag{6-53a}$$

$$W_h^{n+1} = W_h^n - a_2 \nabla \times (W_e^{n+1} + W_e^n) \tag{6-53b}$$

where

$$W_e = \begin{pmatrix} E_x & E_y & E_z \end{pmatrix}^T \tag{6-54a}$$

$$W_h = \begin{pmatrix} H_x & H_y & H_z \end{pmatrix}^T \tag{6-54b}$$

and ∇ should be understood as curl symbol containing difference operators. Inserting Eqn. (6-53b) into Eqn. (6-53a) obtains

$$W_e^{n+1} + b^2 \nabla \times \nabla \times W_e^{n+1} = W_e^n - b^2 \nabla \times \nabla \times W_e^n + 2a_1 \nabla \times W_h^n \tag{6-55}$$

Eqn. (6-55) can be further simplified and written as

$$(I + b^2 D) W_e^{n+1} = (I - b^2 D) W_e^n + 2a_1 \nabla \times W_h^n \tag{6-56}$$

where

$$D = \begin{pmatrix} -D_{2y} - D_{2z} & D_x D_y & D_x D_z \\ D_x D_y & -D_{2y} - D_{2z} & D_y D_z \\ D_x D_z & D_y D_z & -D_{2x} - D_{2y} \end{pmatrix} \tag{6-57}$$

It can be seen that after expanding Eqn. (5-56) with Eqn. (6-57), it is the same as Eqn. (5-19). This method is more general and concise.

6.7.5 Numerical validation

The methods proposed in this Chapter have been coded and run for a 200 by 200 by 200 cells space at Courant numbers up to 10 at mesh density 50 CPWs and 100 CPWs successfully. Fig. 6-2 and Fig.6-3 are the contour map of the electric field

strength E_x with a line source lying along x-coordinate axis using CNCSU method. It shows that the line source does produce a cylindrical wave: in the plane of perpendicular to and cutting through the line source, it is a circle; in the plane parallel to the line source, the equal-amplitude fronts are straight lines. The distorted circle near the source is due to the un-symmetrical behavior of the split updating procedures like ADI [101].

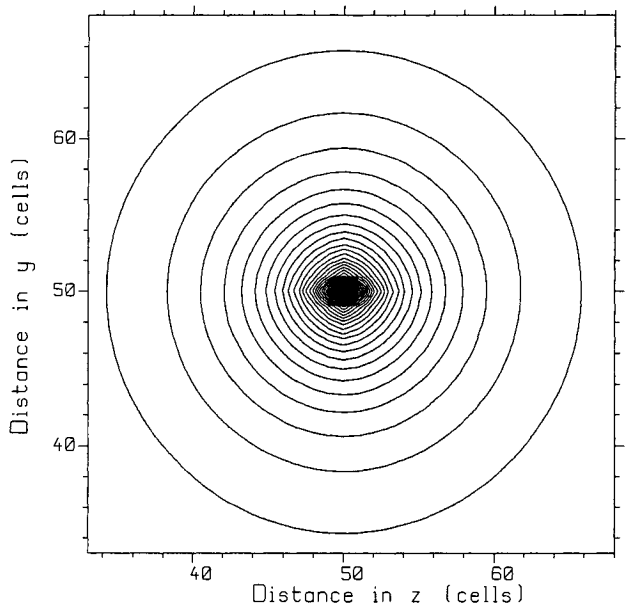

Fig. 6-2 The contour map of the electric field strength E_x of the *y-z* cut.

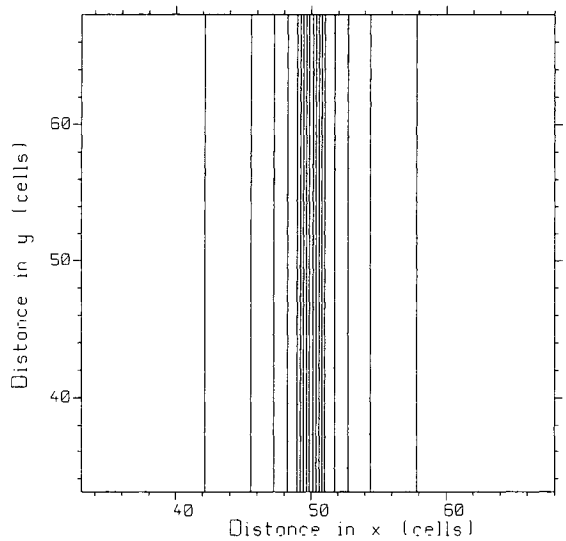

Fig. 6-3 The contour map of the electric field strength E_x of the *x-z* cut.

6.8 Summary

By different approximations, this Chapter proposes several methods in efficient implementing the 3D CN scheme, which are unconditionally stable. The perturbation errors are given, and the amplification factor and numerical dispersion relation for CNDS, CNCSU and CNAFS are derived. The CNDS method has the same numerical dispersion relation as the ADI method. The CNCSU and the CNAFS have much less numerical anisotropy than that of ADI, which is best suited for coefficient-modification technique to reduce the numerical dispersion. The CNDS and the CNAFS can have a little higher computational efficiency for large objects than ADI because they have fewer loops at each time step. CNDS and CNAFS are suitable for parallel computation.

The conditionally-stable methods, including ADI, CN and CN-based FDTD methods are relatively new, their characteristics, potential problems have not been revealed and understood completely. The next Chapter discusses some of the fundamentals for the unconditionally-stable methods.

Chapter 7 Some Fundamental Characteristics of the Unconditionally-Stable FDTD Methods

Explicit FDTD methods have matured through the efforts of many researchers. Thousands of papers, several books [4-6] [10], and chapters in many other text books have been published. The characteristics of explicit FDTD methods have been studied extensively. Even so some papers are still published that reveal some new features and new problems every year. For the recently-developed unconditionally-stable, implicit FDTD methods, such as ADI-FDTD and CN-based FDTD, their characteristics have not been explored fully and understood completely. This chapter will explore some of their fundamental properties.

7.1 Intrinsic Spatial Dispersion

When the discretization is only applied to space and not to time, the result is "semi-discretization". It also has numerical error, called "numerical dispersion of spatial discretization." [128]. This thesis uses the term "intrinsic spatial dispersion" to describe the semi-discretization error, because it is intrinsic in FDTD. In lossless media, the intrinsic spatial dispersion is related to the formulation of the spatial derivatives. The intrinsic spatial dispersion for Yee's FDTD is

$$\frac{u}{c}=\sqrt{\frac{\sin^2(\beta_x\Delta x/2)}{(\beta_0\Delta x)^2}+\frac{\sin^2(\beta_y\Delta y/2)}{(\beta_0\Delta y)^2}+\frac{\sin^2(\beta_z\Delta z/2)}{(\beta_z\Delta z)^2}} \tag{7-1a}$$

Note that the numerical velocity u is also anisotropic [128]. For the optimized methods in Chapter 3, because the difference operators are not conventional simple central differencing, the error in the semi-discretized form can be written as

$$\frac{u}{c} = \sqrt{\sum_{i=x,y,z} \frac{\sin^2(\beta_i \Delta i/2)}{(\beta_0 \Delta i)^2} [w(w_{24} + (1-w_{24})p_{i24}) + (1-w)(w_{na} + (1-w_{na})p_{ina})]^2} \quad (7\text{-}1b)$$

where the weight parameters are given in Chapter 3. It can be seen that such numerical error in spatial discretization does depend on the formulation of the spatial-derivative difference approximation.

For ADI and CN-based methods, let $\Delta t \to 0$, and then the semi-discretized equations are the same as those for Yee' method. Thus they have the same spatial discretization error as the Yee's FDTD. However, after time discretization, the time-step-size related cross terms are introduced from Chapters 4, 5 and 6, which lead to different behaviors of numerical dispersion.

7.2 Non-Physical Values at the Intermediate Time Step

Except for the CNAD and the CNDS methods in the 2D case, all other unconditionally-stable methods, such as ADI, CNDG, CNAFS, and CNCS in 2D and 3D, have to be factorized by adding some higher-order terms in order to be implemented efficiently. The factorized equations must be split into two or even three equations using intermediate step(s). In ADI-FDTD, the intermediate step is associated with n+1/2; other methods do not associate any specific time value and the intermediate time step is denoted with "*". Although some textbooks have pointed out that the field values at the ADI's intermediate time step are non-physical, and are only temporary values to complete a full time step advancement [78], some users of the method do not realize or do not understand this fact, and erroneously use the non-physical values as "true" values in sub-griding schemes. This incorrect use will seriously affect the numerical accuracy. However, in CN-based methods, the intermediate step is part of the marching procedure,

it is not possible in construction to use the erroneous intermediate field values in sub-griding schemes.

7.3 Numerical Errors

The electromagnetic field is a wave which has amplitude and phase. Therefore numerical methods usually have errors both in the amplitude and in the phase. Numerical dispersion and anisotropy are phase errors, and the numerical dissipation and growth as well as numerical loss error introduced in Chapter 9 are amplitude errors. For a wide-band signal, according to the Fourier series, phase error can also cause magnitude error. To evaluate the global accuracy of a method, one can compare the numerical results to analytic results for individual special problems. However, such comparison results are case-dependent. Thus this thesis uses numerical dispersion errors and numerical loss error for evaluation of a method.

For Yee's FDTD in 1D, since its numerical dispersion relation is $\sin^2(\omega\Delta t/2) = s^2 \sin^2(\beta\Delta x/2)$, the "magic time step" of $\Delta t = \Delta x/c$ for $s = 1$ has zero numerical dispersion error [4]. However, since the tangent cannot be cancelled by the sine, it is generally not possible for ADI and CN-based methods to have zero numerical dispersion without modification. To eliminate the numerical dispersion along some specific direction of travel, the coefficient-modification technique can be used as discussed in previous chapters.

Because CN-based methods such as CNDG and CNCS are more isotropic, and even can eliminate numerical anisotropy at one specific time step size for one frequency, the coefficient-modification technique is more efficient and suitable to reduce or eliminate numerical dispersion in all directions of travel.

In Yee's FDTD, *smaller* time step sizes incur *larger* dispersion errors. In contrast, for ADI-FDTD and CN-based methods the numerical dispersion decreases as the time step size becomes smaller.

7.4 Limits on Time Step Size and Numerical Attenuation for ADI-FDTD

As pointed out in Chapter 2, the Nyquist criterion must be obeyed in numerical computation. This section shows that for the unconditionally-stable FDTD methods, the Nyquist criterion follows naturally from the numerically-propagating wave in lossless space without numerical attenuation.

7.4.1 Nyquist limit

Though ADI and CN-based methods are stable for any time step size, there are other limitations than stability. Notice that in the numerical dispersion relations Eqs. (4-5), (4-17), (4-21), (5-5c), (5-8d), (6-51), if the argument of the tangent is π or its integer multiples, the numerical phase constant β must be zero, which means that the numerical wave does not propagate. Numerical experiments have verified this observation. This corresponds to a time step size $\Delta t_1 = 2\pi / \omega$. If $|\tan(\omega\Delta t/2)| > s$, the numerical phase constant β becomes complex, and the wave decays in space. This never happens in physics in lossless material. In the extreme case, the tangent function is infinite if its argument is $\pi/2$ or odd multiples, and the numerical wave decays infinitely fast. The case $\omega\Delta t/2 = \pi/2$ corresponds to a time step size $\Delta t_2 = 2\pi/(2\omega) = 1/(2f)$. The time step size Δt_2 coincides with the Nyquist criterion [131]. Hence there is an upper-bound limit to the usable time step size for ADI and CN-based methods even though they are still stable beyond this limit. This limit is not related to stability. The Nyquist criterion relating the Courant number and mesh density N can be written as

$$s \le N/2 \tag{7-2}$$

Therefore the useable range of the Courant number s is at most up to $N/2$. Beyond this limit it is impossible to recover the signal from a mathematical point of view, and will have numerical attenuation.

7.4.2 Numerical attenuation in 1D

For a deep insight into the time step size limit, take the 1D case as a simple example. In 1D case, all the unconditionally-stable methods in this thesis have the same numerical dispersion relation

$$\tan^2(\omega\Delta t/2) = s^2 \sin^2(\beta\Delta x/2) \tag{7-3}$$

Rewrite Eqn. (7-3) using the Euler formula and express the phase constant as a complex number

$$\cosh(j\beta\Delta x) = \frac{2\tan^2(\omega\Delta t/2)}{s^2} - 1 \tag{7-4a}$$

$$\beta = \beta_r + j\beta_i \tag{7-4b}$$

where β_r is the real phase constant and β_i is the attenuation constant. Eqn. (7-4a) can be written as two equations

$$1 - \cosh(\beta_i\Delta x)\cos(\beta_r\Delta x) = 2\tan^2(\omega\Delta t/2)/s^2 \tag{7-5a}$$

$$\sinh(\beta_i\Delta x)\sin(\beta_r\Delta x) = 0 \tag{7-5b}$$

In (7-5b), either the hyperbolic sine is equal to zero or the sine is zero, or both. If $|\tan(\omega\Delta t/2)| < s$, the attenuation constant β_i is zero, so $\sinh(\beta_i\Delta x)$ is zero. The numerical wave travels without attenuation. When $|\tan(\omega\Delta t/2)| = s$, both $\sinh(\beta_i\Delta x)$ and $\sin(\beta_r\Delta x)$ are zero, and a minimum velocity limit is reached. This is the "transition" point. When

$|\tan(\omega\Delta t/2)| > s$, $\sin(\beta_r \Delta x)$ maintains zero-valued for whatever Courant number, so the following equation can be written

$$\beta_r = \pi / \Delta x \tag{7-6a}$$

From (7-5a), it can be found that

$$\beta_i = \ln\left((2\tan^2(\omega\Delta t/2)/s^2 - 1) \pm \sqrt{(2\tan^2(\omega\Delta t/2)/s^2 - 1)^2 - 1}\right)/\Delta x \tag{7-6b}$$

The above results are similar to that of Yee's "faster than light" propagation in [4]. The $\pm$ sign in (7-6b) indicates that the computed electromagnetic fields may increase or decrease in space. Neither the increasing fields nor the decreasing fields in free space are physical. Taking the minus sign in (7-6b), the wave is attenuated and the attenuation is a function of mesh density N and the Courant number. When the Courant number reaches the Nyquist limit, the wave has infinite attenuation and it cannot travel. The attenuation gives a practical upper-bound limitation to the allowable time step size.

7.4.3 The slowest numerical velocity and the transition mesh density

Eqn. (7-6a) gives the limit for the slowest numerical wave as

$$u_m = 2c / N \tag{7-7}$$

At $|\tan(\omega\Delta t/2)| > s$, the numerical wave travels at the extremely low velocity solved by Eqn. (7-7) for N larger than 2, which is only a function of the mesh density. Table 7-1 gives the velocity limit for various mesh densities.

Table 7-1 The minimum velocity limit and velocity error for various mesh densities.

N	5	10	15	20	25
u_m	0.40c	0.20c	0.13c	0.10c	0.08c
$1-u_m/c$	60%	80%	87%	90%	92%

By analogy with explicit FDTD [4], from Eqn. (7-6b) the transition mesh density where the numerical phase constant changes from a real and a complex number can be found to be

$$N_{transition} = \frac{\pi \quad s}{\tan^{-1}(s)} \tag{7-8}$$

The relationship between the Courant number and the transition mesh density is graphed in Fig. 7-1, labeled as 1D limit. It can be seen that the 1D limit is within the Nyquist limit. If the mesh density is chosen larger than the 1D limit (above the curve), it has no numerical attenuation. For example, if a Courant number of 2 is used, the mesh density must be larger than 5.67 CPW to avoid numerical attenuation.

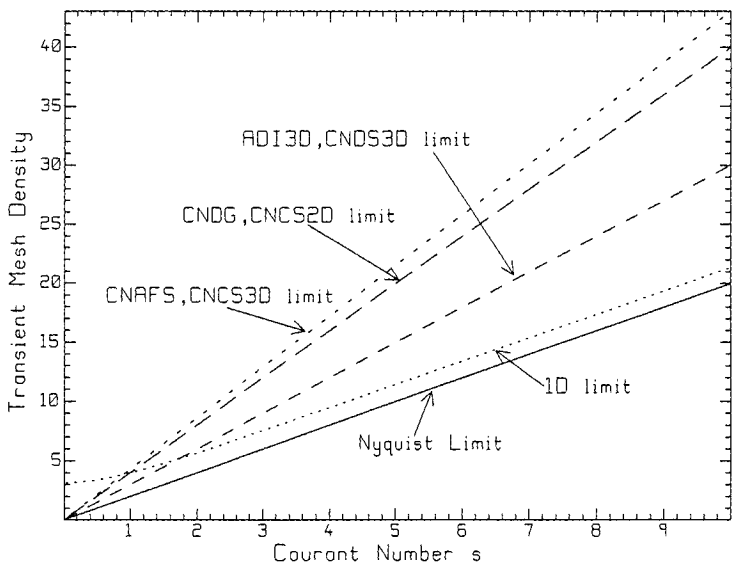

Fig. 7-1 The transition mesh density verses the Courant number.

Knowing this relation is important. It gives information about what frequency cannot be propagated without numerical attenuation. For a given cell size Δx, the mesh density $N = \lambda / \Delta x$ is proportional to the wavelength λ. This transition mesh density corresponds to the highest frequency that can be propagated in the FDTD grid, given by

$$f_{\max} = \frac{c}{N_{transition}\Delta x} = \frac{2s}{N_{transition}} f_{Nyquist} \tag{7-9}$$

where $f_{Nyquist}$ is the frequency at the Nyquist criterion. One should not expect to get information correctly from FDTD simulation data with frequencies higher than Eqn. (7-9), even though the excitation source contains such high frequency. In addition, if the excitation source contains frequencies higher than Eqn. (7-9), then a FDTD simulation will lead to an incorrect result. This explains what happens to the high-frequency energy in the turn-on transient.

For example, for a Courant number of 5, the transition mesh density is 11.44 CPW. If a mesh density is chosen to be smaller than 11.44 CPW, the frequencies higher than $(2*5/11.44)f_{Nyquist} = f_{Nyquist}/1.144$ cannot travel without numerical attenuation.

7.4.4 Faster-than-light speed

From Eqn. (7-7) it can be seen that, if N is smaller than 2, the numerical velocity is larger than the speed of light. But from the spatial Nyquist limit described in Chapter 2, the mesh density cannot be smaller than 2. Thus "faster-than-light" travel cannot occur in 1D unconditionally-stable methods. With similar analysis, the "faster-than-light" travel cannot occur in Yee's 1D FDTD either. The conclusion in [4] comes from the transform of Eqn. (7-7) into $u_m = c/s$ using the relation Eqn. (7-2) $s = N/2$, and suppose $s < 1$, then "faster-than-light" travel is realized. However, this violates the spatial Nyquist criterion.

7.4.5 Courant number

In this thesis, the Courant number has been defined as $s = c\Delta t / \Delta x$. From this expression, several relations can be found. For a given mesh density and Courant number, the time step size is

$$\Delta t = \frac{s}{N}\frac{1}{f} \tag{7-10}$$

Thus, the higher the frequency which is associated with the mesh density, the smaller the time step size. If the time period of a sinusoidal signal for a frequency is $T = 1/f$, then the following expression is valid

$$s = \frac{\Delta t / T}{\Delta x / \lambda} \tag{7-11}$$

The numerator and the denominator stand for the time sampling rate and the spatial sampling rate, respectively. In Yee's FDTD, $s \leq 1$, therefore time is over-sampled compared to the space sampling. For unconditionally-stable methods, the space is over-sampled because the Courant number is often larger than one. For a given Courant number, a coarse mesh density requires large time step size, which leads to large numerical error. Thus unconditionally stable methods are not suitable and not designed for coarse meshes in terms of accuracy. The evaluation of an USM's performance with Yee's method for the same mesh density is inappropriate.

7.5 Time-Step-Size Limits for 2D and 3D Methods

In higher dimensions, the time step size limits along the axes are the same as 1D. Thus this Section only discusses the diagonal direction of travel. Similar to 1D case, for a uniform mesh, the numerical phase constant β_r, the attenuation constant β_i along the diagonals for square mesh can be written as

$$1-\cosh(A\beta_i\Delta x)\cos(A\beta_r\Delta x)=B \tag{7-12a}$$

$$\sinh(A\beta_i\Delta x)\sin(A\beta_r\Delta x)=0 \tag{7-12b}$$

$$A=\begin{cases}\sqrt{2}/2 & 2D\\ \sqrt{3}/3 & 3D\end{cases} \tag{7-12c}$$

$$\beta_i=\ln\left((B-1)\pm\sqrt{(B-1)^2-1}\right)/(A\Delta x) \tag{7-12d}$$

where B depends on the specific FDTD methods, and is given in Table 7-2 in 2D and Table 7-3 in 3D. For CN in 2D and 3D, and for ADI, CNAD and CNDS in 2D, the transition mesh density is the same as the 1D limit, because the numerical velocity along diagonals is always larger than that along axes. For CNDG, CNCS, CNAFS, the numerical velocity along the diagonal can be smaller than that along the axes when the Courant number is larger than the "zero-anisotropy Courant number" [153]. ADI in 3D behaves similarly. Thus they have stricter limit as shown in Fig.7-1.

Table 7-2 Different t0ransition mesh densities and quantities B in 2D

Method	B	$N_{transition}$
CN	$\frac{2\tan^2(\pi s/N)}{s^2}$	$\frac{\pi\ s}{\tan^{-1}(s)}$
ADI,CNAD,CNDS	$\frac{2\left(\sqrt{\tan^2(\pi s/N)+1}-1\right)}{s^2}$	$\frac{\pi\ s}{\tan^{-1}(s)}$
CNDG,CNCS	$\frac{2\left(1\pm\sqrt{1-\tan^4(\pi s/N)}\right)}{s^2\tan^2(\pi s/N)}$	$4s$
YEE	$\frac{2\sin^2(\pi s/N)}{s^2}$	$\frac{\pi\ s}{\sin^{-1}(s)}$

Table 7-3 Different transition mesh densities and quantities B in 3D

Method	B	$N_{transition}$
CN	$\frac{2\tan^2(\pi s/N)}{3s^2}$	$\frac{\pi\ s}{\tan^{-1}(s)}$
ADI, CNAD	$2\frac{3+\tan^2(\pi s/N)\pm\sqrt{9+6\tan^2(\pi s/N)-3\tan^4(\pi s/N)}}{s^2\tan^2(\pi s/N)}$	$3s$
CNAFS	$\frac{2\left(-\frac{a}{4\tan(\pi s/N)}-\frac{1+\tan^2(\pi s/N)}{a\tan(\pi s/N)}-1\pm I\frac{\sqrt{3}}{2}\left[\frac{a}{2\tan(\pi s/N)}-\frac{2(1+\tan^2(\pi s/N))}{a\tan(\pi s/N)}\right]\right)}{s^2}$ $a=\left(4\left(1+\tan^2(\pi s/N)\right)\left(-3*\tan(\pi s/N)+\sqrt{5*\tan^2(\pi s/N)-4}\right)\right)^{1/3}$ $\frac{\pi s}{\tan^{-1}(\sqrt{4/5})}\approx 4.3s$	4.3s
YEE	$\frac{2\sin^2(\pi s/N)}{3s^2}$	$\frac{\pi\ s}{\sin^{-1}(s)}$

For 3D ADI, this transition mesh density is at $3s$, because now the velocity along the diagonal can be smaller than that along the axes. For CNAFS, the limit is stricter than that for CNDG and CNCS. Note that for smaller Courant numbers, if the numerical velocity along the diagonal is larger than that along the axes, the limits for CNDG, CNCS, CNAFS and the 3D ADI are the same as the 1D limit.

The transition point and the numerical velocity are given by

$$\beta_r = \frac{\pi}{A\Delta x} \tag{7-13a}$$

$$u_m = A\frac{2c}{N} \tag{7-13b}$$

The parameter A (which is reciprocal of the square root of the number of the dimensions) can be considered as a "transformer" to transfer higher dimensions to "1D". In higher dimensions, the mesh density along one coordinate axis is N for a square (cubic) mesh.

The equivalent 1D mesh density N_e for N is N/A. Then Eqn. (7-13b) can be re-written as

$$u_m = \frac{2c}{N_e} \tag{7-13c}$$

which is the same as Eqn. (7-7). Thus if N_e reaches the coarsest mesh density --the Nyquist limit of 2, Eqn. (7-13c) indicates that the faster-than-light phenomenon cannot happen along the diagonals. However, the propagation along the axes will have this phenomenon since now $N = AN_e < 2$.

This section stresses that the ADI-FDTD and CN-based methods *do have* an upper-bound limit to the time step size. This upper-bound does not come from the stability requirement as in Yee's FDTD, but arises from the fact that in a lossless medium there must be no attenuation, which leads to a stricter limit than the Nyquist criterion. In reality, the meaningful maximum allowable time step size to be used in the unconditionally-stable FDTD methods should be determined from the desired accuracy, which is much smaller than the upper-bound limit.

7.6 Choosing a Proper Time Step Size for a Desired Accuracy

In solving a practical problem using an USM, the first parameter to be chosen is the mesh density, which is determined by the finest geometrical feature that must be resolved. The second parameter is the desired numerical accuracy, namely the largest numerical dispersion error that can be tolerated. To maintain this desired accuracy defined as

$$p = 1 - \frac{u_m}{c} \tag{7-10}$$

where u_m corresponds to the numerical velocity at the desired accuracy p, the time step size must be chosen properly [146].

7.6.1 Choosing a time step size according to axis dispersion accuracy

For CN and ADI methods, and for other CN-based methods with a Courant number no larger than the zero-anisotropy Courant number, the numerical dispersion error along the axes is the largest along all directions of travel. Thus only the 1D equation (7-2a) will be discussed. Suppose the largest cell size is $\Delta_{\max} = \max(\Delta x, \Delta y, \Delta z)$. For a desired accuracy p and a given mesh density, the time step size can be chosen from [155]

$$\frac{\tan(\omega \Delta t / 2)}{c\Delta t / \Delta_{\max}} = \sin\left(\frac{\beta \Delta_{\max}}{2}\right) \tag{7-11}$$

The minimum numerical velocity is $u_p = \omega / \beta$. Thus combining (7-10) and (7-11) obtains

$$\sin\left(\frac{\pi}{(1-p)N}\right) = \frac{\tan(\pi s / N)}{s} \tag{7-12}$$

where $s = c\Delta t / \Delta_{\max}$ is the Courant number, and $N = \lambda / \Delta_{\max}$ is the minimum mesh density corresponding to a maximum mesh size of $\Delta_{\max}$. Eqn. (7-12) is the relation between the desired dispersion accuracy p, the minimum mesh density and the Courant number. Thus giving a maximum mesh size $\Delta_{\max}$ hence minimum mesh density N, a simple, direct way to determine the time step size to achieve desired dispersion accuracy is to solve Eqn. (7-12) for the Courant number s using a simple root-finding algorithm, and then evaluate the time step size as

$$\Delta t = s\Delta_{\max} / c \tag{7-13}$$

Note that the time step size for the desired dispersion accuracy is only a function of the largest mesh size, and not the aspect ratio [155] for ADI and other CN-based methods when the Courant number is smaller than the zero-anisotropy Courant number.

Fig. 7-2 graphs the Courant number s as a function of the mesh density N for various dispersion accuracy values. Solving (7-12) shows an almost-linear relation between the Courant number and the mesh density.

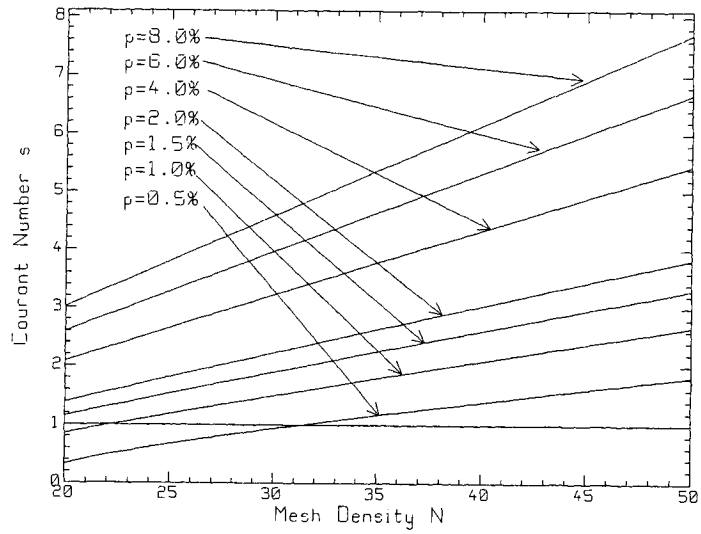

Fig. 7-2 Relation between the Courant number s and the mesh density N, for various dispersion accuracies p.

7.6.2 For methods with negative anisotropy

For CNDG, CNCS and CNAFS, when the Courant number is larger than the zero-anisotropy Courant number, the numerical dispersion error is the largest along the diagonals. In this case, the time step size must be chosen from the diagonals. The time step sizes for 2D and 3D square mesh can be found from the following equations

$$2\sin^2\left(A\frac{\pi}{(1-p)N}\right)=\left(1+s^4\sin^4(A\frac{\pi}{(1-p)N})\right)\frac{\tan^2(\pi s/N)}{s^2} \qquad (7\text{-}14a)$$

$$3\sin^2\left(A\frac{\pi}{(1-p)N}\right)=\left(1+3s^4\sin^4(A\frac{\pi}{(1-p)N})+s^6\sin^6(A\frac{\pi}{(1-p)N})\right)\frac{\tan^2(\pi s/N)}{s^2} \qquad (7\text{-}14b)$$

where A is given in Eqn. (7-12c). For non-square mesh, the Courant number is a function of aspect ratio, and can be determined numerically from the numerical dispersion relation by a similar method to that given above.

7.6.3 Relative Courant number

In practice, to evaluate whether an unconditionally-stable method is more efficient than Yee's FDTD for the same dispersion accuracy, the relative Courant-Friedrich-Levy number (RCFLN) is used [158], defined as the ratio of the time step size for the unconditionally-stable method to the Courant time step size limit for the Yee's FDTD

$$RCFLN=\frac{\Delta t}{\Delta t_{\max}^{Yee}} \qquad (7\text{-}15)$$

where the Courant limit for Yee's FDTD is

$$\Delta t_{\max}^{Yee}=\frac{\Delta_{\max}}{cR} \qquad (7\text{-}16)$$

and the aspect ratio of the mesh R is defined as

$$R=\sqrt{(\Delta_{\max}/\Delta x)^2+(\Delta_{\max}/\Delta y)^2+(\Delta_{\max}/\Delta z)^2} \qquad (7\text{-}17)$$

It can be shown that the *RCFLN* can be expressed in terms of the Courant number and the aspect ratio as

$$RCFLN=sR \qquad (7\text{-}18)$$

This *RCFLN* is the same as the CFLN defined in [146]. For an USM to be more efficient than Yee's, the *RCFLN* must be 4 to 5 in the 3D case, to compensate for the larger number of arithmetic operations in the USM. *RCFLN* of 4 to 5 corresponds to a Courant

number of 2.3 to 2.9. Thus the relation (7-18) can be used to evaluate whether an USM is suitable for a specific problem in terms of over-all CPU time consumption.

7.7 Intrinsic Temporal Numerical Dispersion

In Yee's FDTD, because of the Courant constraint, the time step size is limited by the mesh size. Therefore numerical dispersion disappears if an infinitely-fine mesh is used. In contrast, ADI and CN-based methods have no such constraint. As the mesh size decreases to zero, the time step size does not need to go to zero to maintain stability. Thus for a given time step size Δt, there will be some numerical dispersion, no matter how small the mesh size. The dispersion at zero mesh size is called "intrinsic temporal numerical dispersion" (ITND) [153-154] due to time discretization.

7.7.1 1D case

From the 1D dispersion equation (7-2), letting Δx go to zero leads to

$$\tan^2(\omega\Delta t/2) = (c\beta\Delta t/2)^2 \tag{7-19}$$

The numerical phase constant is $\beta = \tan(\omega\Delta t/2)/(c\Delta t/2)$, thus the intrinsic temporal numerical dispersion is expressed as

$$\frac{u}{c} = \frac{\omega\Delta t/2}{\tan(\omega\Delta t/2)} \tag{7-20}$$

Fig. 7-3 graphs the ITND according to (7-20) at four different frequencies f, $2f$, $3f$ and $4f$. The x-axis is the relative time step size $\Delta t/\Delta t_{Nyquist}$, where $\Delta t_{Nyquist} = 1/2f$ is the Nyquist limit for frequency f. The ITND can be quite large. For example, when the relative time step size is chosen to be $\Delta t/\Delta t_{Nyquist} = 0.10$, the numerical dispersion is $2\pi/\beta = \pi/20/\tan(\pi/20) = 0.9918$ at frequency f and the velocity error is about 0.82%. At the Nyquist limit $\Delta t_{4f}/\Delta t_{Nyquist} = 0.25$ for the signal of $4f$, the numerical

wave speed is zero and the wave does not travel even though it is not discretized in space. From Eqn. (7-4b) it can be seen that the attenuation is infinite at the Nyquist limit. It should be pointed out that the ITND is the same for higher-dimensional CN scheme as it is in 1D. However, ADI and other CN-based methods have different behaviors in higher dimensions.

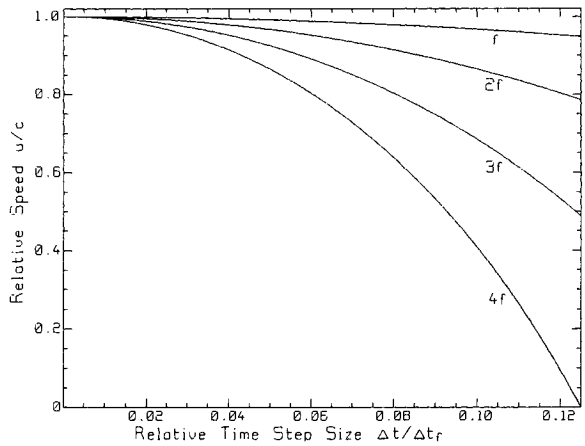

Fig. 7-3 The intrinsic temporal numerical dispersion at different frequencies.

7.7.2 2D case

In the 2D case, as the cell sizes goes to zero, the intrinsic temporal numerical dispersion can be derived from the numerical dispersion relation as [153]

$$\frac{u}{c}=\frac{\omega\Delta t/2}{\tan(\omega\Delta t/2)}\sqrt{\frac{1+\sqrt{1+g(\tan(\omega\Delta t/2)\sin 2\phi)^2}}{2}} \tag{7-21a}$$

$$g=\begin{cases}1 & ADI,CNAD,CNDS\\ 0 & CN\\ -\tan^2(\omega\Delta t/2) & CNDG,CNCS\end{cases} \tag{7-21b}$$

Numerical calculations using Eqn. (7-21) show that the temporal intrinsic numerical dispersion as a function of the direction of travel is similar to the dispersion curve at finite cell size. From Eqn. (7-21) it can be seen that the relative velocity is not a function of direction of travel for the CN method. Therefore CN's anisotropy is zero at the zero

mesh size limit. But for ADI and CNDG, there is anisotropy, as shown in Fig. 7-4 and Fig. 7-5. This anisotropy is termed the "intrinsic temporal anisotropy." Note that the ADI's anisotropy is about 30 times larger than that of CNDG at the time step size of one-tenth of the Nyquist time step size limit.

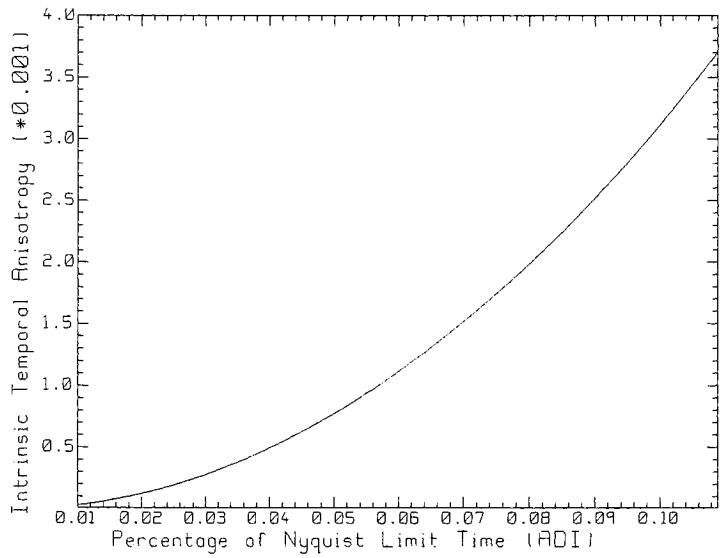

Fig. 7-4 Intrinsic temporal anisotropy with zero mesh size for 2D ADI and CNAD.

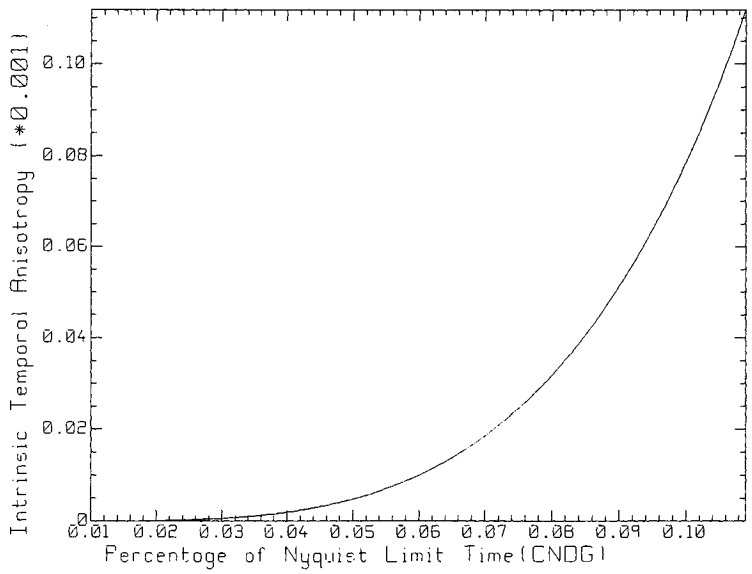

Fig. 7-5 Intrinsic temporal anisotropy with zero mesh size for 2D CNDG and CNCS.

7.7.3 3D case

The equation to be solved for the 3D intrinsic temporal numerical dispersion is

$$\begin{aligned}&\tan^2(\omega\Delta t/2)+(\omega\Delta t/2u_r)^6\sin^4(\theta)\cos^2(\theta)\sin^2(\varphi)\cos^2(\varphi)\\&=(\omega\Delta t/2u_r)^2+g(\omega\Delta t/2u_r)^4\left(\sin^4(\theta)\sin^2(\varphi)\cos^2(\varphi)+\sin^2(\theta)\cos^2(\theta)\right)\end{aligned} \tag{7-22a}$$

$$u_r=\frac{\beta_0}{\beta}=\frac{u}{c} \tag{7-22b}$$

where g_1 and g_2 are given in Eqn. (6.52). In practice, the mesh density is never zero, and the numerical dispersion is larger than the intrinsic temporal numerical dispersion, therefore the intrinsic temporal numerical dispersion is the fundamental limit of accuracy of USMs.

7.8 Classification of Problems Suitable for Unconditionally-Stable Methods

Problems suitable for the USMs have been identified partly in [84] [103] [155], and can be classified as: 1) entire domain problem, previous called the ADI class [155], and 2) hybrid problem where a coarse mesh encloses a fine mesh, in which the coarse mesh uses an explicit FDTD method, and the fine mesh uses an USM. Each of the two categories can be further classified into two sub-categories according to the aspect ratio of the mesh: that with very large aspect ratio and that with relatively small aspect ratio, such as 5. For problems with large aspect ratio, to save CPU time, the "hybrid implicit-explicit" FDTD scheme [147] can be used, where the explicit operation is applied on the axis with coarse mesh density, the implicit operation is applied on the axis with fine mesh density. The time step size is determined from the explicit method. The USMs proposed in this thesis are better suited in efficiency for problems where the aspect ratio is not too larger, though they can be used in such cases.

For the entire domain problem with not-too-large aspect ratio, the time step size chosen for a desired accuracy is described above. However, for hybrid problems, to avoid time interpolation between the coarse mesh and the fine mesh, the time step size is usually chosen to be the same for both meshes, which is limited by the Courant constraint, and according to the accuracy requirement of the coarse mesh enclosing the fine mesh for USMs. In this case, the accuracy in the fine mesh is determined from the chosen time step size. The next section will describe the accuracy of the hybrid problems.

Keep in mind that USMs are favored over their explicit counterparts for some problems, in which the time step size necessary for procuring a required temporal accuracy may be significantly larger than that dictated by the explicit stability condition, or the explicit methods consume so much CPU time that it is prohibitively long.

7.9 Accuracy of Hybrid Schemes for Sub-Gridding

A hybrid scheme uses a sub-grid of fine cells embedded within a coarse mesh of Yee cells [88]. Such a "hybrid scheme" updates the coarse mesh with Yee's FDTD and the fine sub-grid with a USM. In [88], ADI-FDTD is used. When Yee's FDTD advances one time step, ADI-FDTD advances only one sub-step using the Yee time step size, and the non-physical intermediate values are incorporated to update the Yee cell space. After updating the Yee space, the ADI sub-grid is updated with the 2^{nd} sub-step. But the erroneous intermediate values are propagated into the Yee grid. A better approach is to advance the ADI sub-grid using half the Yee time step for both sub-steps, each time the Yee grid is updated. In the following we assume that the ADI-FDTD uses the same time step size Δt as the Yee's grid for a complete update cycle of ADI.

Define $m = \Delta x_{Yee} / \Delta x_{ADI}$ as the coarse-to-fine ratio, where Δx_{Yee} is the coarse mesh size for explicit method, and Δx_{ADI} is the fine mesh size for ADI-FDTD. Using the same time step size for both grids, it can be shown that the Courant number s_{ADI} for ADI-FDTD is related to the Courant number s_{Yee} of Yee's FDTD as

$$s_{ADI} = m s_{Yee} \tag{7-23}$$

It can be seen that the higher the coarse-to-fine ratio, the larger the Courant number for ADI-FDTD. Analysis shows that as long as the time step size is chosen so that the Yee grid is stable, ADI-FDTD cannot reach the time step size limit shown in Fig. 7-1. Thus, the ADI grid will not experience any numerical attenuation.

Since the accuracy of the ADI-FDTD increases with a finer mesh size, it is expected that for a given Yee mesh size, the accuracy of the ADI sub-grid improves as the coarse-to-fine ratio increases. Fig. 7-6 shows the relative velocity in the ADI sub-grid as a function of the ADI mesh density for Yee mesh densities of 10 and 20 cells per wavelength, and for two Courant numbers in each case. For reference, the relative velocity in the Yee coarse grid is shown for $s_{Yee} = 0.707$ (corresponding to the 2D Courant limit) at the top of the figure. If the Yee coarse grid uses a mesh density of 10 cells per wavelength and Courant number 1.0, then the velocity error in the ADI sub-grid is about 3.3% with mesh density 100 (m=10). If a smaller time step size is used in the coarse grid, for example, $s_{Yee} = 0.707$, the velocity error in the fine grid decreases to 1.7% at mesh density 100. But note that the smaller time step size decreases the velocity accuracy in the coarse grid. If the Yee grid is finer with N=20, then the velocity error in the ADI-FDTD can be made less than 1.0%.

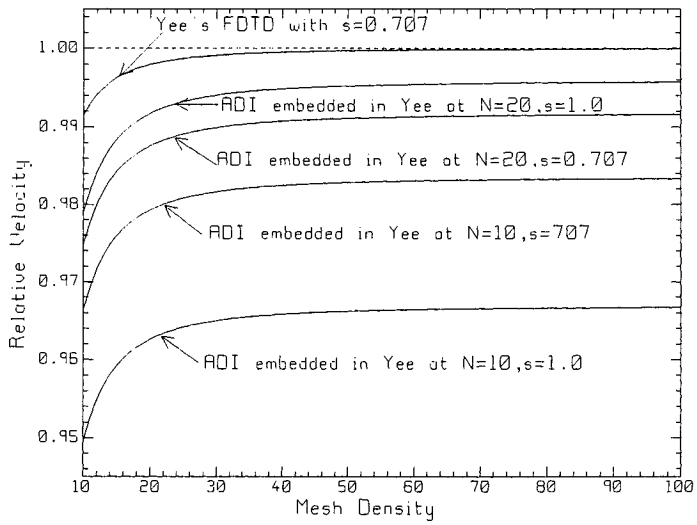

Fig. 7-6 Numerical dispersion of ADI-FDTD when hybridizing with Yee-FDTD.

It can be seen from Fig. 7-6 that the velocity increases sharply for ADI-FDTD from mesh density 10 to 30 and then increases slowly from 30 to 100. Though the accuracy of ADI-FDTD increases with a finer sub-grid, the CPU time and memory requirements will be larger. Little increase in accuracy in the ADI sub-grid is achieved for coarse-to-fine ratios greater than 5, which is close to the intrinsic temporal dispersion error. In Fig. 7-6, the velocity error improves about 0.05% from mesh density 50 (m=5) to 100 (m=10) for the ADI-FDTD in all cases. This analysis assumes the same time step size for a complete update cycle of ADI as in the Yee coarse grid. If the time step size is chosen in the Yee grid for high accuracy, then to have the same accuracy in the ADI sub-grid as in the Yee coarse grid, a smaller time step size would be needed in the ADI sub-grid. Thus a time interpolation between the Yee coarse mesh and the ADI fine mesh is required. The above discussion is effective for other CN-based methods provided that the time step size does not exceed the zero-anisotropy time step size discussed in Chapter 5.

7.10 Amplification Polynomial and Amplification Factors

It has been noted that the amplification polynomials for the Yee's FDTD, ADI-FDTD, and the CN-based methods in this Thesis can be written in a unified form as

$$P(\xi) = (1-\xi)^{m_1}(p_0 + p_1\xi + p_2\xi^2)^{m_2} \tag{7-24}$$

where

$$m_1 = \begin{cases} 0 & 1D \\ 1 & 2D \\ 2 & 3D \end{cases} \qquad m_2 = \begin{cases} 1 & 1D \\ 1 & 2D \\ 2 & 3D \end{cases} \tag{7-25}$$

The parameter m_1 is the number of stationary solution with $\xi = 1$, and the parameter m_2 is the number of non-stationary solutions, which have given by

$$\xi = \frac{-p_1 \pm J\sqrt{4p_0p_2 - p_1^2}}{2p_2} \tag{7-26}$$

The condition of stability is derived from the quantity inside the square-root by requiring that $4p_0p_2 \geq p_1^2$. The magnitude of the amplification factor under the stability condition is $|\xi| = p_0 / p_2$: if $p_0 > p_2$, there is numerical growth; if $p_0 < p_2$, there is numerical dissipation; if $p_0 = p_2$, there is neither growth nor dissipation, thus the method is strictly non-dissipative. Therefore, if the constant p_0 and the quadratic coefficient p_2 are the same, the scheme should be stable. It can be shown that the amplification polynomial of the full update cycle for ADI-FDTD fulfills this condition, thus is strictly non-dissipative for the overall ADI-FDTD method. However its individual sub-marching procedures behave growth or dissipative because of $p_0 \neq p_2$.

From Chapter 2 it is known that the number of solutions of the amplification polynomial is related to the number of dimensions, which is $m_1 + 2m_2$. But what is the

role of the stationary mode solution $\xi = 1$? Some authors refer it as static eigenmode with eigenfrequency $\omega = 0$ [148]. This may mislead people to believe that the FDTD method has static field inherently. Analysis shows that m_1 corresponds to the number of *longitudinal* field components, and m_2 corresponds to the number of *transverse* field components. Thus it is understood that the solutions reveal the fundamental property of the electromagnetic wave: the transverse wave. The Fourier analysis is the approximate result of a far field cylindrical or spherical wave, which has no longitudinal components. Thus the stationary solution corresponds to the longitudinal mode. In addition, the parameter m_1 also corresponds to the number of Divergence Theorems, and the total number of solutions corresponds to the total number of the electromagnetic field components.

In addition, the amplification factor shows the time evolution part of $e^{\pm J\omega t}$. Therefore, a stable FDTD scheme must have the amplification polynomial written in the form of Eqn. (7-24), particularly the m_2 term. Otherwise, the solution that corresponds to the non-stationary modes will have two different frequencies, which is not correct, since the Fourier method uses a monochromatic wave. However, in the literature, it has been found that there are some papers that obtained two different frequencies (TAP, Vol.51, No.7, pp.1615-1622, July, 2003). From this example it can be seen that we need to understand more mathematically more about FDTD.

7.11 Summary

This Chapter has discussed several fundamental characteristics of the unconditionally-stable methods. For some choices of cell and time step sizes, the mesh will not propagate the numerical wave. Although the time step size in USMs is not

limited by the cell size, it is governed by the Nyquist criterion, and the numerical dispersion relation implies a time-step-size limit which is more strict and smaller than the Nyquist limit. The intrinsic spatial dispersion is the same as Yee's FDTD. However, different from Yee's FDTD, an USM has the intrinsic temporal numerical dispersion, and thus there is an accuracy limit as the mesh density becomes finer. Problems suitable for USMs have been classified as the entire domain problems and hybrid problems. The time step size for a desired accuracy is given for the entire domain problems. In the hybrid scheme, the same time step size is used in both the Yee grid and the fine sub-grid. In the fine sub-grid a numerical wave can always be propagated without numerical attenuation, but the sub-grid will have a larger dispersion error than for the Yee coarse grid.

A stable FDTD scheme must have the amplification polynomial that can be written in Eqn. (7-24). The number of stationary solutions of the amplification polynomial is the number of longitudinal field components, and the number of non-stationary solutions of the amplification polynomial is the number of transverse field components, which reveals the transverse property of the electromagnetic wave.

The "faster-than-light" travel does not occur neither in Yee's FDTD nor in the unconditionally-stable methods introduced in this thesis when the mesh density along each axis is above the Nyquist limit. Note that the transition mesh density given in Tables 7.2 and 7.3 should be compared to Eqn. (7-8) and use the largest to determine the faster-than-light phenomenon.

Chapter 8 Discretization Error Quantification With the Fourier Method

From previous Chapters, it can be seen that numerical dispersion error only measures phase error. This Chapter proposes a method that can quantify the discretization error of the difference equations, which is a measure of the magnitude error, based on the Fourier mode. In addition, this chapter reviews all the methods proposed previously and casts them into more general formulation if possible. For simplicity, only the lossless case is considered.

8.1 Introduction

In order to overcome some limitations in the traditional Yee's FDTD method, many alternative schemes have been proposed. To compare numerical schemes, truncation error analysis using the Taylor Series and numerical dispersion are often used. The former indicates the order of accuracy of a scheme. The latter shows how the numerical wave velocity differs from physical speed. A third approach is to quantify the amplitude of the error numerically with some benchmark problems. However, these measures are sometimes insufficient to quantify, distinguish and compare different methods. For example, bench-mark problems may be case-dependent. Both ADI-FDTD and CNDS (CNAP) are (2, 2) schemes and have the same numerical dispersion in the 2D case. However, the actual magnitude error of the two methods is different, which will be shown in this Chapter.

The Fourier method has been used in amplitude error analysis for parabolic problems [76] [127]. It has been found [161] that the Fourier method can also be used to analyze

the order of accuracy of an FDTD scheme, to characterize the discretization error and compare quantitatively the accuracy among different FDTD schemes. Such error is independent of time, space and frequency.

The "transfer function" concept introduced in [149] is used in this thesis to quantify and compare schemes, and to give insight into the spectral resolution of the differential or difference operators. Define the spatial transfer function of a Fourier mode e^{-Jkx} and the temporal transfer function of a Fourier mode $e^{-J\omega t}$ as

$$T(k) = -\frac{F(e^{-Jkx})}{Je^{-Jkx}} \tag{8-1a}$$

$$T(\omega) = \frac{F(e^{J\omega t})}{Je^{J\omega t}} \tag{8-1b}$$

where k is the physical phase constant, to distinguish it from the numerical phase constant β previously used, and F is the differential or the finite-difference operator. Note that the minus sign in Eqn. (8-1a) is intended to obtain positive value of the transfer function. The basic idea of the Fourier method is to find the difference between the transfer functions of the continuous-differential equations and of the discrete-difference counterparts using one Fourier mode. Since Maxwell's Equations are a system of equations, a matrix expression is required to connect the relations.

8.2 Theoretical Plane-Wave Matrix

In Maxwell's Equations given in Eqn. (2-6), E and H have different units. For convenience, the electric field components need to be normalized to have the unit of the magnetic field. For a TE_z wave in a linear, isotropic, non-dispersive and lossless material, the Curl Equations can be written in matrix form as

$$\partial W = 0 \tag{8-2}$$

$$\partial = \begin{pmatrix} c^{-1}\partial_t & 0 & -\partial_y \\ 0 & c^{-1}\partial_t & \partial_x \\ -\partial_y & \partial_x & c^{-1}\partial_t \end{pmatrix} \qquad W = \begin{pmatrix} E_{hx} \\ E_{hy} \\ H_z \end{pmatrix} \tag{8-3}$$

where ∂ is the partial differential operator matrix, $E_{hx} = E_x / \eta$, $E_{hy} = E_y / \eta$, and $\eta = \sqrt{\mu / \varepsilon}$ is the intrinsic impedance. In the finite difference method, the partial differential operator matrix ∂ is approximated with the finite difference operator matrix D. The Fourier method evaluates the discretization (or truncation) error matrix $\partial - D$ with proper scaling [161].

For a general solution, the discretization error is space- and time- dependent. To remove such dependence, the Fourier method uses the transfer function [149] with the fundamental plane wave solution of the PDEs or FDEs, as defined in Eqn. (8-1). The 2D plane wave or the Fourier mode is given by

$$\psi = \psi_0 e^{J\omega t} e^{-J(k_x x + k_y y)} \tag{8-4}$$

where $k_x = k\cos\phi$, $k_y = k\sin\phi$, and $k = \sqrt{k_x^2 + k_y^2}$ is the physical phase constant. Thus, for PDEs, the transfer functions are $T_{0,t} = (d\psi / dt)/(J\psi) = \omega$, $T_{0,x} = -(d\psi / dx)/(J\psi) = k_x$, and $T_{0,y} = -(d\psi / dy)/(J\psi) = k_y$. The subscript "0" stands for theory. For FDEs, they are $T_t = D_t(\psi)/(J\psi)$, $T_x = -D_x(\psi)/(J\psi)$ and $T_y = -D_y(\psi)$ $(J\psi)$. For example, for Yee's FDTD, $T_t = \omega w_t$, $T_x = k w_x$ and $T_y = k w_y$, where $w_t = \sin(\omega\Delta t/2)/(\omega\Delta t/2)$, $w_x = \sin(k_x\Delta x/2)/(k\Delta x/2)$ and $w_y = \sin(k_y\Delta y/2)/(k\Delta y/2)$.

8.2.1 Theoretical coefficient matrix

For the TE_z wave, insert Eqn. (8-4) into Maxwell's Curl Equations given by Eqn. (8-2) and eliminate the common factor $J\psi$ to obtain the following linear homogeneous equations

$$A_0 W^T = 0 \tag{8-5a}$$

$$W^T = \begin{bmatrix} E_{hx0} & E_{hy0} & H_{z0} \end{bmatrix}^T \tag{8-5b}$$

$$A_0 = \begin{bmatrix} c^{-1}T_{0,t} & 0 & T_{0,y} \\ 0 & c^{-1}T_{0,t} & -T_{0,x} \\ T_{0,y} & -T_{0,x} & c^{-1}T_{0,t} \end{bmatrix} \tag{8-5c}$$

where A_0 is the theoretical coefficient matrix. By setting the determinant of A_0 to zero, the dispersion relation $k = \omega c^{-1}$ can be obtained.

8.2.2 Normalized plane-wave decomposition matrix

It has been shown that the plane wave must also be decomposed in order to quantify the error properly [161]. From Fig. 8-1 it can be shown that the plane wave decomposition is realized with a decomposition matrix T_p

$$\begin{pmatrix} E_{xh0} & E_{yh0} & H_{z0} \end{pmatrix}^T = T_p \begin{pmatrix} H_{z0} & H_{z0} & H_{z0} \end{pmatrix}^T \tag{8-6a}$$

$$T_p = \begin{pmatrix} -\sin\phi & 0 & 0 \\ 0 & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8-6b}$$

where ϕ is the direction of wave travel. Thus, the theoretical plane-wave matrix is

$$A_{p,0} = A_0 T_p = \begin{pmatrix} -k_y & 0 & k_y \\ 0 & k_x & -k_x \\ -k_y s_\phi & -k_x c_\phi & k \end{pmatrix} \tag{8-7}$$

where $c_\phi = \cos\phi$ and $s_\phi = \sin\phi$. In the following, we define $c_t = \cos(\omega\Delta t/2)$ and $s_t = e^{-J\omega\Delta t/2}$ to shorten the expressions.

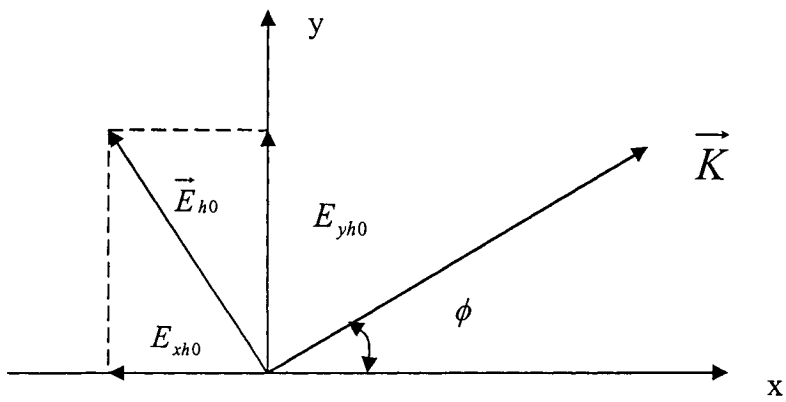

Fig. 8-1 Decomposition of a plane wave.

8.3 Plane Wave Matrix for Explicit FDTD Methods

From Chapter 3, a general explicit discretization for Maxwell's Equations can be formulated as

$$c^{-1}D_t E_{hx}^{n+1/2} - D_{H,y}H_z^{n+1/2} = 0 \tag{8-8a}$$

$$c^{-1}D_t E_{hy}^{n+1/2} + D_{H,x}H_z^{n+1/2} = 0 \tag{8-8b}$$

$$c^{-1}D_t H_z^n - D_{E,y}E_{hx}^n + D_{E,x}E_{hy}^n = 0 \tag{8-8c}$$

where $D_{E,\bullet}$ and $D_{H,\bullet}$ stand for the spatial difference operators acting on the E field and the H field components, where the operator formulas may not be the same for E and H, such as in [34] [41] [44]. Inserting the plane wave Eqn. (8-4) into the explicit FDTD update equations Eqs. (8-8), the following coefficient matrix can be obtained

$$A_{EX} = \begin{bmatrix} c^{-1}T_t & 0 & T_{H,y} \\ 0 & c^{-1}T_t & -T_{H,x} \\ T_{E,y} & -T_{E,x} & c^{-1}T_t \end{bmatrix} \tag{8-9}$$

By setting the determinant of matrix A_{EX} equal to zero, a general numerical dispersion relation for explicit FDTD methods with 2nd order accuracy in time can be obtained as

$$c^{-2}T_t^2 = T_{E,x}T_{H,x} + T_{E,y}T_{H,y} \tag{8-10}$$

Eqn. (8-10) can be used to derive all the numerical dispersion relations given in Chapter 3 and those in [34] [40] [44].

For an explicit scheme having 2nd order temporal accuracy, T_t is the same as the Yee's FDTD, but the spatial transfer functions depend on the formulation of the spatial-difference approximations. For Yee's FDTD, the coefficient matrix is

$$A_{Yee} = k\begin{bmatrix} w_t & 0 & w_y \\ 0 & w_t & -w_x \\ w_y & -w_x & w_t \end{bmatrix} \tag{8-11}$$

The general plane-wave matrix for the difference equations Eqs. (8-8) is

$$A_{p,EX} = A_{EX}T_p = \begin{pmatrix} -c^{-1}T_t s_\phi & 0 & T_{H,y} \\ 0 & c^{-1}T_t c_\phi & -T_{H,x} \\ -T_{E,y}s_\phi & -T_{E,x}c_\phi & c^{-1}T_t \end{pmatrix} \tag{8-12}$$

Thus the plane wave matrix for Yee's FDTD is

$$A_{p,Yee} = k\begin{bmatrix} -w_t s_\phi & 0 & w_y \\ 0 & w_t c_\phi & -w_x \\ -w_y s_\phi & -w_x c_\phi & w_t \end{bmatrix} \tag{8-13}$$

The plane wave matrices for other explicit methods can be written similarly.

8.4 Plane Wave Matrices for CN-Based Methods

In this section, the spatial difference operators for CN-based methods use 2nd - order central difference formula, so the transfer functions will be written as specific instead of the general for easy understanding.

The Crank-Nicolson scheme [151] [157] is balanced at the time step n+1/2 at both the LHS and RHS as described in Chapter 4. Other CN-based methods proposed in this

thesis are formulated by adding some higher-order terms as discussed in Chapter 5. For the original CN scheme, with similar derivation to the explicit methods, the coefficient matrix can be written as [161]

$$A_{CN} = k\begin{bmatrix} w_t & 0 & w_y c_t \\ 0 & w_t & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{bmatrix} \tag{8-14}$$

Eqn. (8-14) differs from Yee's method in Eqn. (8-11) by the factor $c_t = \cos(\omega\Delta t/2)$. The plane wave matrix can be obtained as

$$A_{p,CN} = k\begin{bmatrix} -w_t s_\phi & 0 & w_y c_t \\ 0 & w_t c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{bmatrix} \tag{8-15}$$

For other CN-based methods, one can derive the plane wave matrices from their update equations. However, such results are complicated and do not reveal explicitly the fact that they are modifications of the CN scheme. For clarity, here we use the formulation that combines the CN scheme with the higher-order terms added to individual CN-based methods given in Chapter 5. With some manipulation, the coefficient matrices are listed in Table 8-1 and the plane wave matrices in Table 8-2 for CNDG, CNDS (CNAP) and CNCS. For CNCS methods, the matrices are almost the same except that the locations of the perturbation errors are different.

In comparison to the CN and the CN-Based methods, ADI-FDTD has complex error terms due to unbalanced time marching scheme, and is complicated.

Table 8-1 Coefficient matrices for CNDG, CNDS (CNAP) and CNCS.

CNDG	$A_{p,CNDG} = k\begin{pmatrix} -s_\phi w_t & 0 & w_y c_t \\ 0 & c_\phi w_t & -w_x c_t \\ -s_\phi w_y c_t & -c_\phi w_x c_t & w_t(1+b^4k^4w_x^2w_y^2) \end{pmatrix}$
CNDS (CNAP)	$A_{CNDSX} = k\begin{pmatrix} w_t & b^2k^2w_xw_yw_t & w_y c_t \\ 0 & w_t & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{pmatrix}$ $A_{CNDSY} = k\begin{pmatrix} w_t & 0 & w_y c_t \\ b^2k^2w_xw_yw_t & w_t & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{pmatrix}$
CNCS-EI	$A_{CNEIX} = k\begin{pmatrix} w_t & 0 & w_y c_t \\ -b^4k^4w_xw_y^3w_t & w_t(1+b^4k^4w_x^2w_y^2) & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{pmatrix}$ $A_{CNEIY} = k\begin{pmatrix} w_t(1+b^4k^4w_x^2w_y^2) & -b^4k^4w_x^3w_yw_t & w_y c_t \\ 0 & w_t & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{pmatrix}$
CNCS-IE	$A_{CNCSIEX} = k\begin{pmatrix} w_t & -b^4k^4w_xw_y^3w_t & w_y c_t \\ 0 & w_t(1+b^4k^4w_x^2w_y^2) & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{pmatrix}$ $A_{CNIEY} = k\begin{pmatrix} w_t(1+b^4k^4w_x^2w_y^2) & 0 & w_y c_t \\ -b^4k^4w_x^3w_yw_t & w_t & -w_x c_t \\ w_y c_t & -w_x c_t & w_t \end{pmatrix}$

Table 8-2 Plane wave matrices for CNDG, CNDS (CNAP) and CNCS.

CNDG	$A_{p,CNDG} = k\begin{pmatrix} -s_\phi w_t & 0 & w_y c_t \\ 0 & c_\phi w_t & -w_x c_t \\ -s_\phi w_y c_t & -c_\phi w_x c_t & w_t(1+b^4k^4w_x^2w_y^2) \end{pmatrix}$
CNDS (CNAP)	$A_{p,CNDSX} = k\begin{pmatrix} -w_t s_\phi & b^2k^2w_xw_yw_tc_\phi & w_y c_t \\ 0 & w_t c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{pmatrix}$ $A_{p,CNDSY} = k\begin{pmatrix} -w_t s_\phi & 0 & w_y c_t \\ -b^2k^2w_xw_yw_ts_\phi & w_t c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{pmatrix}$
CNCS-IE	$A_{p,CNCSIEX} = k\begin{pmatrix} -w_t s_\phi & 0 & w_y c_t \\ b^4k^4w_xw_y^3w_ts_\phi & w_t(1+b^4k^4w_x^2w_y^2)c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{pmatrix}$ $A_{p,CNCSIEY} = k\begin{pmatrix} -w_t(1+b^4k^4w_x^2w_y^2)s_\phi & 0 & w_y c_t \\ b^4k^4w_x^3w_yw_ts_\phi & w_t c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{pmatrix}$
CNCS-EI	$A_{p,CNCSEIX} = k\begin{pmatrix} -w_t s_\phi & 0 & w_y c_t \\ b^4k^4w_xw_y^3w_ts_\phi & w_t(1+b^4k^4w_x^2w_y^2)c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{pmatrix}$ $A_{p,CNCSEIY} = k\begin{pmatrix} -w_t(1+b^4k^4w_x^2w_y^2)s_\phi & -b^4k^4w_x^3w_yw_tc_\phi & w_y c_t \\ 0 & w_t c_\phi & -w_x c_t \\ -w_y c_t s_\phi & -w_x c_t c_\phi & w_t \end{pmatrix}$

8.5 Plane Wave Matrix for ADI-FDTD

ADI-FDTD is a two-step method and cannot use the same procedure in the preceding Sections for the one-step methods. To use the Fourier method quantifying its discretization error, the two sub-steps must be combined into one-step as in Eqs. (4-24) and (4-25). The matrices can be written as

$$A_{ADI}=k\begin{pmatrix} w_t(1+b^4k^4w_x^2w_y^2) & 0 & w_yc_t \\ -2b^4k^4w_xw_y^3s_t/(\omega\Delta t) & w_t(1+b^4k^4w_x^2w_y^2) & -w_xc_t+2b^3k^3w_xw_y^2s_t/(\omega\Delta t) \\ w_yc_t & -w_xc_t & w_t(1+b^4k^4w_x^2w_y^2) \end{pmatrix} \tag{8-16a}$$

$$A_{p,ADI}=\begin{pmatrix} -w_t(1+b^4k^4w_x^2w_y^2)s_\phi & 0 & w_yc_t \\ 2b^4k^4w_xw_y^3s_t/(\omega\Delta t)s_\phi & w_t(1+b^4k^4w_x^2w_y^2)c_\phi & -w_xc_t+2b^3k^3w_xw_y^2s_t/(\omega\Delta t) \\ -w_yc_ts_\phi & -w_xc_tc_\phi & w_t(1+b^4k^4w_x^2w_y^2) \end{pmatrix} \tag{8-16b}$$

8.6 Error Matrices and RMS Errors

The difference between the theoretical plane wave matrix and the numerical plane wave matrix reflects the accuracy of the numerical method used. The discretization error for the plane wave is defined as

$$e_D=kH_{z0}e=k\left(A_{0,p}-A_p\right)\left(H_{z0}\quad H_{z0}\quad H_{z0}\right)^T \tag{8-17}$$

where e is defined as the time- and space- independent relative error as

$$e=\left(e_1\quad e_2\quad e_3\right)^T=\Delta A\left(1\quad 1\quad 1\right)^T \tag{8-18}$$

where e_1 , e_2 and e_3 are the normalized discretization errors of the three FDEs or update equations for 2D-FDTD methods, respectively. Thus the quantification of the discretization error lies on the evaluation of the error matrix ΔA. For explicit FDTD methods, the error matrix is

$$\Delta A=k^{-1}\left(A_{0,p}-A_p\right)=\begin{pmatrix} -(kc)^{-1}\left(T_{0,t}-T_t\right)\sin\phi & 0 & -k^{-1}\left(T_{0,y}-T_{H,y}\right) \\ 0 & (kc)^{-1}\left(T_{0,t}-T_t\right)\cos\phi & k^{-1}\left(T_{0,x}-T_{H,x}\right) \\ k^{-1}\left(T_{0,y}-T_{E,y}\right)\sin\phi & k^{-1}\left(T_{0,x}-T_{E,x}\right)\cos\phi & (kc)^{-1}\left(T_{0,t}-T_t\right) \end{pmatrix} \tag{8-19}$$

From Eqn. (8-23) it can be seen that, in order to reduce the numerical error, the numerical transfer functions must be as close to the theoretical transfer functions as possible.

To be efficient, the relative RMS (root-mean-square) error defined as

$$e_{rms}=\sqrt{|e_1|^2+|e_2|^2+|e_3|^2} \tag{8-20}$$

can be used to unitize the three individual errors in Eqs. (8-18) and (8-19). The errors are functions of the mesh and time step sizes, as well as ϕ, hence are anisotropic.

Based on the analysis mentioned above, the following error matrices for explicit FDTD methods in this thesis can be obtained. For Yee's FDTD, the error matrix is

$$\Delta A_{Yee} = \begin{pmatrix} -(1-w_t)\sin\phi & 0 & (\sin\phi - w_y) \\ 0 & (1-w_t)\cos\phi & -(\cos\phi - w_x) \\ -(\sin\phi - w_y)\sin\phi & -(\cos\phi - w_x)\cos\phi & (1-w_t) \end{pmatrix} \tag{8-21}$$

For the optimized methods in Chapter 3, the error matrix is

$$\Delta A_w = \begin{pmatrix} -(1-w_t)\sin\phi & 0 & (\sin\phi - w_y p_{wy}) \\ 0 & (1-w_t)\cos\phi & -(\cos\phi - w_x p_{wx}) \\ -(\sin\phi - w_y p_{wy})\sin^2\phi & -(\cos\phi - w_x p_{wx})\cos\phi & (1-w_t) \end{pmatrix} \tag{8-22}$$

where p_{wx} and p_{wy} are method-related, and can be found from Chapter 3. It can be seen that, in order to reduce discretization error, for a given time marching method, such as Yee's leap-frog scheme, the spatial transfer functions due to spatial discretization must be close to the temporal transfer function.

Table 8-3 gives the error matrices for various unconditionally-stable methods. Once the error matrices are known from Eqs. (8-21), (8-22) and Table 8-3, the discretization errors of different schemes can be evaluated quantitatively.

8.7 Discretization Errors for Various FDTD Methods

This section demonstrates the properties of the discretization errors for various FDTD methods.

8.7.1 Explicit methods

Fig. 8-2 shows the individual discretization errors of Yee's FDTD for a mesh density of 10 CPW at the Courant limit. It can be seen that equations of E_{hx} and E_{hy} have similar

Table 8-3 Error matrices for the unconditionally-stable methods.

CN	$\Delta A_{CN}=\begin{pmatrix} -(1-w_t)s_\phi & 0 & (s_\phi-w_y c_t) \\ 0 & (1-w_t)c_\phi & -(c_\phi-w_x c_t) \\ -(s_\phi-w_y c_t)s_\phi & -(c_\phi-w_x c_t)c_\phi & (1-w_t) \end{pmatrix}$
ADI	$\Delta A_{ADI}=\begin{pmatrix} -\left(1-w_t(1+b^4k^4w_x^2w_y^2)\right)s_\phi & 0 & s_\phi-w_y c_t \\ -2b^4k^4w_x w_y^3 s_t/(\omega\Delta t)s_\phi & \left(1-w_t(1+b^4k^4w_x^2w_y^2)\right)c_\phi & -\left(c_\phi-w_x c_t+2b^3k^3w_x w_y^2 s_t/(\omega\Delta t)\right) \\ -\left(s_\phi-w_y c_t\right)s_\phi & -\left(c_\phi-w_x c_t\right)c_\phi & \left(1-w_t(1+b^4k^4w_x^2w_y^2)\right) \end{pmatrix}$
CNDS CNAD	$\Delta A_{CNDSX} = \begin{pmatrix} -(1-w_t)s_\phi & -b^2k^2w_x w_y w_t c_\phi & (s_\phi-w_y c_t) \\ 0 & (1-w_t)c_\phi & -(c_\phi-w_x c_t) \\ -(s_\phi-w_y c_t)s_\phi & -(c_\phi-w_x c_t)c_\phi & (1-w_t) \end{pmatrix}$ $\Delta A_{CNDSY} = \begin{pmatrix} -(1-w_t)s_\phi & 0 & (s_\phi-w_y c_t) \\ b^2k^2w_x w_y w_t s_\phi & (1-w_t)c_\phi & -(c_\phi-w_x c_t) \\ -(s_\phi-w_y c_t)s_\phi & -(c_\phi-w_x c_t)c_\phi & (1-w_t) \end{pmatrix}$
CNDG	$\Delta A_{CNDG} = \begin{pmatrix} -(1-w_t)s_\phi & 0 & (s_\phi-w_y c_t) \\ 0 & (1-w_t)c_\phi & -(c_\phi-w_x c_t) \\ -(s_\phi-w_y c_t)s_\phi & -(c_\phi-w_x c_t)c_\phi & \left(1-w_t(1+b^4k^4w_x^2w_y^2)\right) \end{pmatrix}$
CNCS- IE	$\Delta A_{CNCSIEX} = \begin{pmatrix} -(1-w_t)s_\phi & b^4k^4w_x w_y^3 w_t c_\phi & (s_\phi-w_y c_t) \\ 0 & \left(1-w_t(1+b^4k^4w_x^2w_y^2)\right)c_\phi & -(c_\phi-w_x c_t) \\ -(s_\phi-w_y c_t)s_\phi & -(c_\phi-w_x c_t)c_\phi & (1-w_t) \end{pmatrix}$ $\Delta A_{CNCSIEY} = \begin{pmatrix} -\left(1-w_t(1+b^4k^4w_x^2w_y^2)\right)s_\phi & 0 & (s_\phi-w_y c_t) \\ -b^4k^4w_x^3 w_y w_t s_\phi & (1-w_t)c_\phi & -(c_\phi-w_x c_t) \\ -(s_\phi w_y c_t)s_\phi & -(c_\phi-w_x c_t)c_\phi & (1-w_t) \end{pmatrix}$

discretization errors but with 90° shift. However, H_z and the RMS errors have a cosine-like variation w.r.t. the direction of travel. All errors are near zero along the diagonal because this specific Courant time step size. Thus the diagonal has the highest accuracy, the same as the numerical dispersion. The maximum error for the three field components are the same, about 0.008. The RMS error behaves like the numerical

dispersion error. In other words, the numerical dispersion error is a kind of "RMS" error.

Fig. 8-3 and Fig. 8-4 graph the discretization errors for IOM and NAI (Chapter 3) optimized at 10 CPW, at mesh density 10 CPW and their time step size limits, respectively. For both methods, E_x has a sine-like and E_y has a cosine-like discretization errors w.r.t the direction of travel, with a maximum error about 0.006, about 3/4 of the error of Yee's method. Notice that H_z is almost isotropic with an error of 0.006. The RMS error is highly isotropic, which is expected because IOM is isotropic.

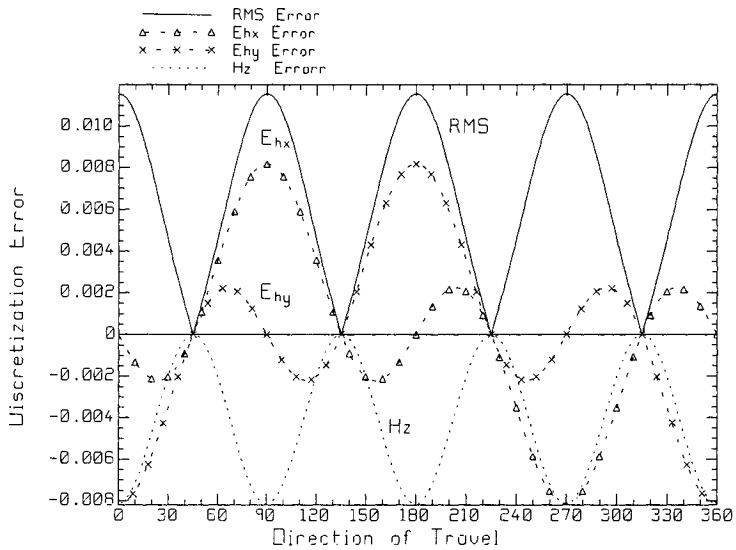

Fig. 8-2 Discretization error of Yee's FDTD with mesh density 10 CPW at the Courant limit.

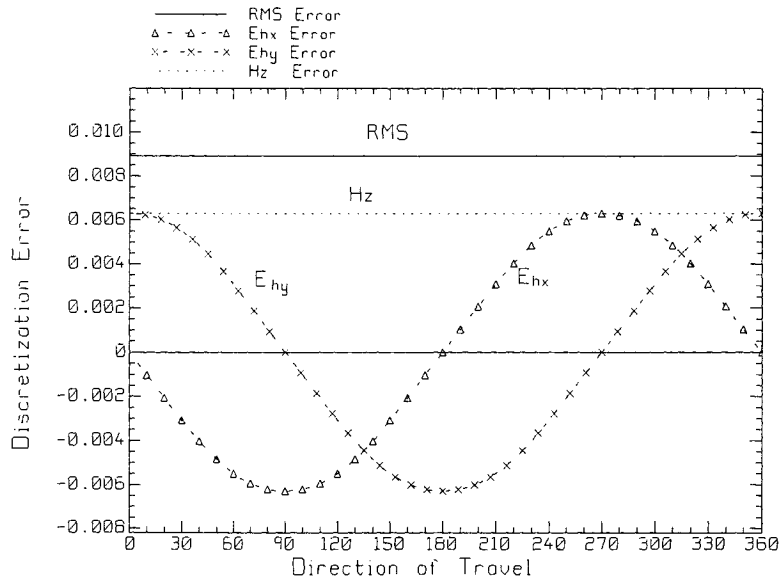

Fig. 8-3 Discretization error of IOM with mesh density 10 CPW at its time step limit $s = 0.848 s_c$.

NAI is a better isotropic method since it has smaller discretization error than IOM. The maximum error of the individual field components is less than 0.005. Again H_z and RMS error are isotropic.

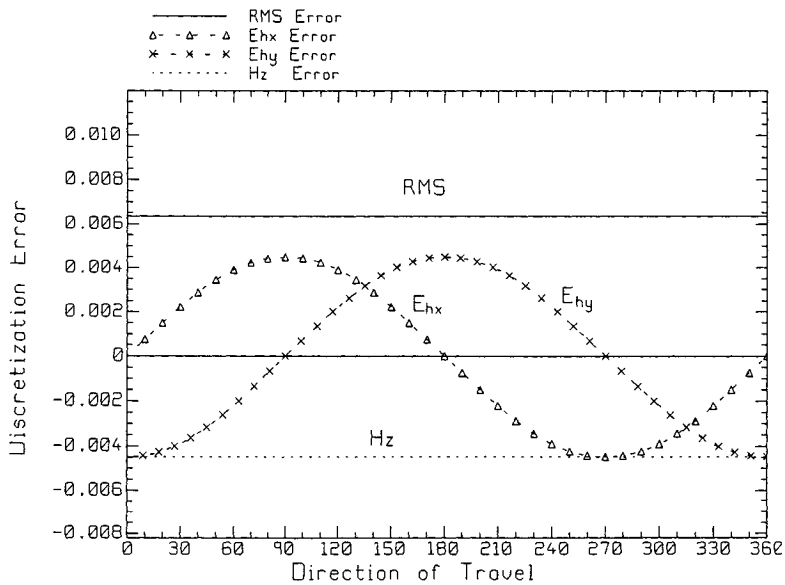

Fig. 8-4 Discretization error of NAI with mesh density 10 CPW at its time step size limit $s = 1.2042 s_c$.

Fig. 8-5 shows the discretization errors for the most accurate explicit method in this thesis, the NA24 method. The maximum error for E_x and E_y is about 2.4×10^{-5}, about 300 times smaller than Yee's FDTD. Notice that H_z has a maximum error of 3×10^{-7}, and is almost isotropic. It can be seen that NA24 has not only a smaller numerical dispersion error, but also much a smaller discretization error. In contrast, the methods in [40] [44] have similar numerical dispersion error as NAI but larger discretization errors similar to Yee's FDTD, because they have used large stencils only for the electric field components or magnetic field component, but not both.

8.7.2 Unconditionally-stable methods

Fig. 8-6 illustrates the RMS error for the unconditionally-stable methods of CN, CNDS (CNAD), ADI and CNCS (CNDG) of Chapter 5. The discretization errors for CNDG and various CNCS methods are indistinguishable since their perturbation errors are 4^{th} –order accurate in space. For clarity only one of the curves is shown. It can be seen that CN and CNCS(CNDG) are more isotropic than ADI, as expected. In most directions of travel, the RMS errors for CNDS and ADI are smaller than those of the more isotropic methods, but over some small angular sectors they have larger RMS errors. Fig. 8-7 gives the maximum RMS errors for CN, CNDS, CNCS(CNDG) and ADI as a function of the Courant number. As the time step size increases, the maximum error increases. However, further data analysis shows that at small time step sizes, the CNDS has a larger maximum error than other methods (see Fig. 8-6), but as time step size increases, its maximum error becomes smaller than those for ADI and CNCS. For example, at mesh density 50 CPW, its maximum error begins to be smaller than ADI's for Courant number larger than 4.5.

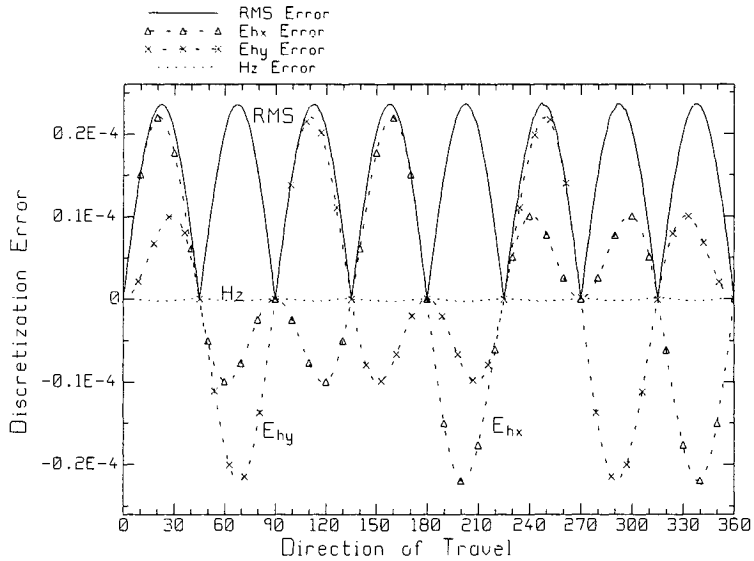

Fig. 8-5 Discretization error of NA24 with mesh density 10 CPW at the Courant limit.

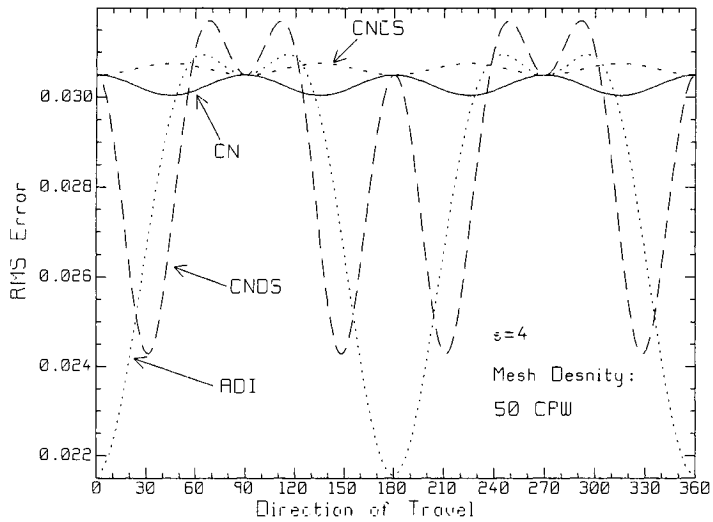

Fig. 8-6 RMS discretization errors of various unconditionally-stable methods with mesh density 50 CPW at the Courant number of 4.

8.8 Discussion

The discretization error discussed above provides information on which field component is more isotropic and which direction of travel is the most accurate. In addition, it can be used to analyze the order of accuracy, and suggest how to reduce the errors, as discussed below.

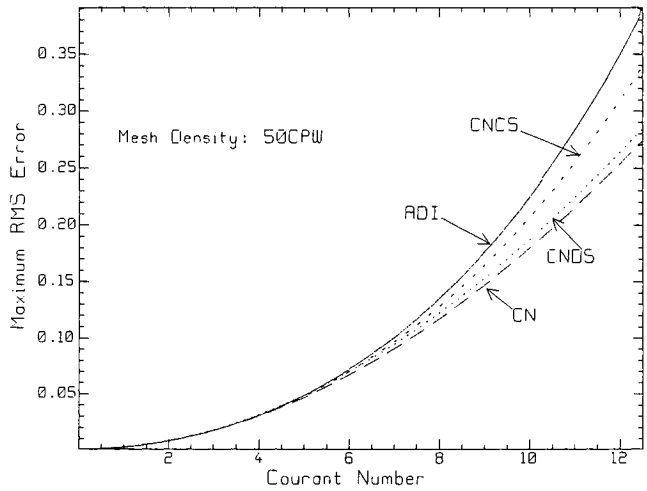

Fig. 8-7 RMS errors of CN, CNDS(CNAD),CNCS and ADI at mesh density 50 CPW.

A. The orders of accuracy of a scheme

From the error matrix, the orders of accuracy of a scheme can be determined easily by expanding the terms as Taylor's Series. Since $1-w_t = O(\Delta t^2)$, $s_\phi - w_x = O(\Delta x^2)$ and $c_\phi - w_y = O(\Delta y^2)$, the Yee's FDTD is a (2,2) scheme. For ADI and CN-based methods, since the lowest orders are $s_\phi - w_y c_t = s_\phi - w_y + 2\sin^2(\omega\Delta t/2) = O(\Delta y^2) + O(\Delta t^2)$ and $c_\phi - w_x c_t = O(\Delta x^2) + O(\Delta t^2)$, the ADI and CN-based methods are also (2,2) accurate.

B. Reduction of discretization errors

From the error matrices it can be seen that in the 1D case, if the "magic time step size" [4] $\Delta t = \sqrt{\mu\varepsilon}\Delta x$ is used, Yee's scheme has neither numerical dispersion error nor discretization error, because it has $w_t = w_x$ or $w_t = w_y$.

It can also be seen that, in order to reduce the discretization error, using a fine mesh is effective for Yee's FDTD because the discretization error is a function of the difference of the transfer functions of the temporal and the spatial derivatives. Since the temporal transfer function is a constant for a given time step size, finer mesh makes w_x and w_y

closer to a constant but still a function of travel direction. This is where the anisotropy comes from.

The optimized methods are another alternative for reducing discretization error. By using large computational stencils, the spatial transfer functions are improved. Thus to reduce the numerical errors in FDTD, the key issue is to design a kind of "spatial filter" with a nearly linear transfer function for a wide bandwidth, and smaller variation w.r.t direction of travel.

8.9 Conclusion

With the fundamental plane wave solution of Maxwell's Equations, the analytical expressions of the discretization errors have been derived for the explicit Yee's FDTD and the implicit FDTD methods described in previous Chapters. This method for error analysis can be used to determine the order of accuracy of a scheme and to compare the accuracy quantitatively between different schemes. The latter is especially important at the development stage of a scheme, which may have the same order of accuracy as another scheme, or even have the same numerical dispersion.

The discretization errors provide more information than the numerical dispersion. For example, it can show which electromagnetic field component has smaller discretization error and along which direction. This may be useful for some particular applications.

In addition, the plane wave matrices can use the numerical phase constant from the numerical dispersion relation, resulting to almost the same the discretization errors as those using the theoretical phase constant. However, using the theoretical phase constant is much simpler and the method can be used for all cases even when the numerical dispersion relation is not known. This method can be extended to the 3D case.

Chapter 9 Numerical Dispersion and Numerical Loss in Lossy Media

Most analysis of the numerical dispersion inherent in FDTD assumes that the material is lossless for simplicity. Although in practice lossy dielectrics and conductors are often encountered in the application of FDTD, the numerical dispersion for lossy materials is seldom investigated in theory or by numerical experiment. This Chapter will show that there is numerical loss in FDTD applied to lossy materials, and the numerical dispersion is also different from lossless case.

9.1 Introduction

The literature in FDTD has neglected the role of loss constant in analyzing the numerical dispersion in the lossy case. For lossy materials, Reference [123] presents the stability analysis of FDTD for a time-average scheme and a time-forward scheme, gives the numerical dispersion relations. Reference [120] analyzes the stability in solving the second-degree wave equation. But both do not include explicitly the loss constant in the Fourier analysis method. Reference [121] generalizes the analysis of stability and numerical dispersion in discrete-convolution FDTD. Recently Reference [124] extends the analysis to the (2, 4) scheme and compares numerical dispersion for (2, 2) and (2, 4) schemes theoretically, using a complex phase constant, but does not mention the loss constant. Reference [122] addresses the numerical dispersion in homogeneous and inhomogeneous waves, and accounts for the loss constant, but omits the σE term in Ampere's Law. No direct comparison between numerical experiment and theory has been reported to validate the dispersion relation in lossy materials.

A correct numerical dispersion relation is required for the understanding and evaluation of an FDTD scheme. For example, the accuracy of the near-to-far zone transformation can be improved if the dispersion relation is known and accounted for [140]. In order to get a correct numerical dispersion relation, both "numerical loss" and conductivity have to be introduced. However, because the magnitude of the electromagnetic fields decreases with distance in space, obtaining numerical loss and dispersion from numerical experiments in higher dimensions with proper accuracy is difficult. Thus this thesis will only validate numerical dispersion and numerical loss in the 1D case for Yee's FDTD, ADI-FDTD and CN-FDTD. In the 2D case, the analysis is provided.

9.2 Numerical Dispersion and Numerical Loss for 1D Yee's FDTD

The 1D update equations for Yee's leap-frog FDTD in a linear, isotropic, lossy and non-dispersive medium are [4]

$$E_y^n(i) = a_0 E_y^{n-1}(i) - a_1 \frac{H_z^{n-1/2}(i+1/2) - H_z^{n-1/2}(i-1/2)}{\Delta x} \tag{9-1a}$$

$$H_z^{n+1/2}(i+1/2) = H_z^{n-1/2}(i+1/2) - a_2 \frac{E_y^n(i+1) - E_y^n(i)}{\Delta x} \tag{9-1b}$$

where the coefficients are $a_0 = (1-\sigma\Delta t/2\varepsilon)/(1+\sigma\Delta t/2\varepsilon)$, $a_1 = (\Delta t/\varepsilon)/(1+\sigma\Delta t/2\varepsilon)$ for "time-average scheme" [120], and $a_2 = \Delta t/\mu$; and ε, μ, σ are the permittivity, permeability and conductivity of the material, respectively. Since the time-average scheme has 2nd-order accuracy in time, and other schemes have lower-order accuracy, this thesis will only discuss various FDTD methods with the time-average scheme.

A theorem has been proved [69] that ensures Yee FDTD in lossy materials is stable within the Courant time-step-size limit. Thus the stability issue will not be addressed

here. From physics, it is known that in a lossy medium the real-valued phase constant must be replaced by a complex-valued propagation constant γ. In addition, it is well known that Yee's FDTD does not give rise to numerical dissipation or growth in time; thus the numerical plane wave can be written as

$$\psi = \psi_0 \exp(J\omega n\Delta t)\exp(-\gamma i\Delta x) \tag{9-2}$$

where $\gamma = \alpha + J\beta$ is the numerical propagation constant, α is the loss constant, and β is the phase constant. Because of discretization, the phase constant β and loss constant α in the computational domain (the difference equations) are generally different from their physical values.

9.2.1 Numerical dispersion and numerical loss relation

To obtain the relation between numerical dispersion and loss, substituting Eqn. (9-2) into Eqn. (9-1) results in the following equations

$$\begin{bmatrix} \left(a_0\xi^{-1/2} - \xi^{1/2}\right) & 2e^{j\varphi}a_1 \sinh(\gamma\Delta x/2)/\Delta x \\ 2a_2 \sinh(\gamma\Delta x/2)/\Delta x & e^{j\varphi}(\xi^{-1/2} - \xi^{1/2}) \end{bmatrix}\begin{bmatrix} E_{y0} \\ H_{z0} \end{bmatrix} = 0 \tag{9-3}$$

where φ is the phase difference between the electric and magnetic fields associated with the conductivity. The amplification polynomial is

$$P(\xi) = \xi^2 - \left(1 + a_0 + r_{Yee}^2\right)\xi + a_0 \tag{9-4}$$

where $r_{Yee} = 2a_1a_2 \sinh(\gamma\Delta x/2)/\Delta x$. It can be seen that the φ term does not appear in the polynomial because of cancellation. Later discussion will omit this term. Setting the polynomial to zero, and expanding the equation in the real and imaginary parts obtains

$$(1+a_0)\sin^2(\omega\Delta t/2) = a_1a_2[1-\cosh(\alpha\Delta x)\cos(\beta\Delta x)]/\Delta x^2 \tag{9-5a}$$

$$(1-a_0)\sin(\omega\Delta t) = 2a_1a_2 \sinh(\alpha\Delta x)\sin(\beta\Delta x)/\Delta x^2 \tag{9-5b}$$

By substituting the constants a_0, a_1 and a_2 into the above equations, the numerical dispersion and loss relation can be obtained as

$$\frac{\sin^2(\omega\Delta t/2)}{(c\Delta t)^2} = \frac{\cosh(\alpha\Delta x)\sin^2(\beta\Delta x/2) - \sinh^2(\alpha\Delta x/2)}{\Delta x^2} \tag{9-6a}$$

$$\sigma\eta\frac{\sin(\omega\Delta t)}{c\Delta t} = 2\frac{\sinh(\alpha\Delta x)\sin(\beta\Delta x)}{\Delta x^2} \tag{9-6b}$$

where $\eta = \sqrt{\mu/\varepsilon}$ is the intrinsic impedance of the corresponding lossless material. Note that the conductivity σ is embedded in the update equations (9-1), and is equal to its physical value. Therefore, Eqn. (9-6) indicates that the numerical phase constant and numerical loss constant do differ from their physical values, and they are coupled.

If σ goes to zero, Eqn. (9-6b) is identical by zero and Eqn. (9-6a) becomes the numerical dispersion equation in lossless case [4]. As the mesh becomes infinitely fine, the time step size must tend towards to zero because of the Courant limit, and then Eqn. (9-6) becomes the theoretical relations as

$$\omega^2 = c^2(\beta_0^2 - \alpha_0^2) \tag{9-7a}$$

$$\sigma\mu\omega = 2\alpha_0\beta_0 \tag{9-7b}$$

where α_0 is the physical loss, and β_0 is the physical phase constant.

In one-dimensional lossless media, Yee's FDTD has a "magic" time step size [4] where the numerical velocity is equal to the physical velocity, resulting in zero numerical dispersion. However, for lossy media, there is no such "magic" time step size. This can be seen by setting $\Delta t = \Delta x / c$ in Eqn. (9-6), which gives

$$\sin^2(\omega\Delta x/(2c)) = \cosh(\alpha\Delta x)\sin^2(\beta\Delta x/2) - \sinh^2(\alpha\Delta x/2) \tag{9-8a}$$

$$\sigma\eta\sin(\omega\Delta x/c)\Delta x = 2\sinh(\alpha\Delta x)\sin(\beta\Delta x) \tag{9-8b}$$

Thus, neither the numerical phase constant nor the numerical loss constant is the same as their physical values in Eqn. (9-7) for any time step size, and are still a function of the conductivity. If a very fine mesh is used, or if the conductivity is very small, the numerical velocity and loss can be quite close to their physical values.

9.2.2 Numerical verification

To demonstrate that Eqn. (9-6) predicts the behavior of the numerical wave correctly in the computational domain, numerical experiments were done with a 1D mesh of 500 cells. A frequency of 300 MHz was used. The mesh density was varied using 10, 20, 30, and 40 CPW in the lossy material. Courant numbers $s = c\Delta t / \Delta x$ of 0.25, 0.50, 0.75, and 1 were used. Conductivities σ of 1, 2, 3, and 4 mS/m were tested, with the permittivity and permeability equal to those of free space. Despite the fact that the physical velocity in the lossy material is different from $c = \sqrt{1/\mu\varepsilon}$, we can still evaluate the numerical dispersion as u/c as in a lossless material. The velocity in lossy media is smaller than that in the corresponding lossless medium, so $u/c < 1$.

Fig. 9-1 shows the relative numerical velocity *u/c* as a function of the mesh density for a conductivity σ=1 mS/m, for the four different Courant numbers. For comparison, the numerical velocity in the corresponding lossless material is shown as a dashed line. For small Courant numbers and coarse mesh densities, the error in the numerical velocity is large, and for the Courant limit, the error is the smallest in both cases. In Fig. 9-1 the results from numerical experiments are also shown with small circles, and they agree quite well with the theoretical prediction of Eqn. (9-6).

Fig.9-2 shows the numerical loss α predicted by Eqn. (9-6), and the results from numerical experiments (small circles) with good agreement. The inconsistency at $s = 1$ is

due to the very small attenuation, which causes the extraction of the loss constant inaccuracy. It can be seen that the numerical loss is always larger than its physical value. Both the numerical dispersion and numerical loss approach their physical values as the mesh becomes finer.

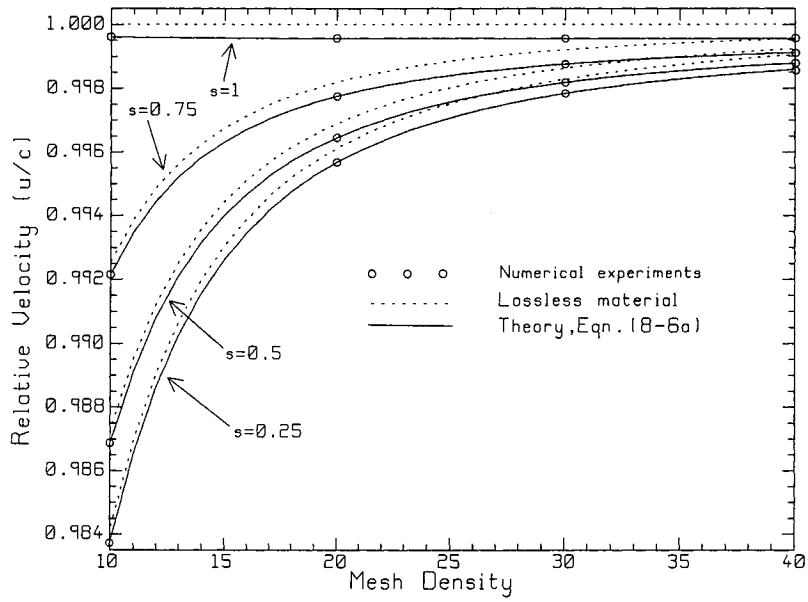

Fig. 9-1 Numerical dispersion for $\sigma = 1$ mS/m at 300 MHz.

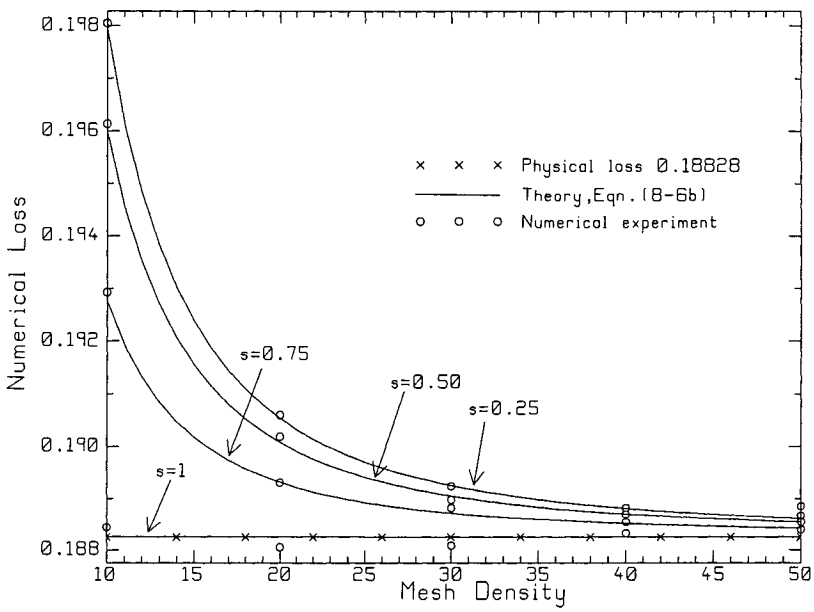

Fig. 9-2 Numerical loss for $\sigma = 1$ mS/m at 300 MHz.

9.2.3 Numerical dispersion and loss over a wide-bandwidth

To understand how the conductivity influences the numerical velocity and numerical loss, Fig.9-3 graphs the numerical velocity percent error, NVPE=$100(u_p - u)/u_p$, where $u_p = \omega/\beta_0$ is the physical velocity in the medium. Fig.9-4 shows the numerical loss percentage error, NLPE=$100(\alpha_0 - \alpha)/\alpha_0$, as a function of the conductivity for typical mesh density 10 cells per wavelength at frequencies of 300 MHz and 3GHz. The magnitude of the numerical velocity percentage error increases as conductivity increases. At 300 MHz, for s=1 in Fig. 9-3, NVPE is negative and increases in magnitude with conductivity, and asymptotically approaches –3.7%. For s=0.707, NVPE is positive for small conductivity, then crosses zero and increases in magnitude as the conductivity increases. For very large conductivity, the NVPE approaches –3.7%. Negative NVPE means the numerical velocity is larger than the physical velocity, which does not occur in the lossless case. In Fig. 9-4, the NLPE is always negative at both frequencies, so the numerical loss is larger than its physical value. For s=1 and 0.707, the NLPE increases in magnitude asymptotically as the conductivity increases. However, for s=0.5, the NLPE decreases and approaches –2.97% with large conductivity. In general the numerical loss is smaller when a fine mesh is used or when a *larger* time-step-size within its Courant limit is used.

In lossy media, the physical velocity and loss constant are both functions of frequency. As the frequency increases, the physical velocity becomes faster and the physical loss becomes larger for a given conductivity. The numerical loss and velocity are also functions of frequency. Comparing the curves for 300 MHz and 3 GHz in Fig.

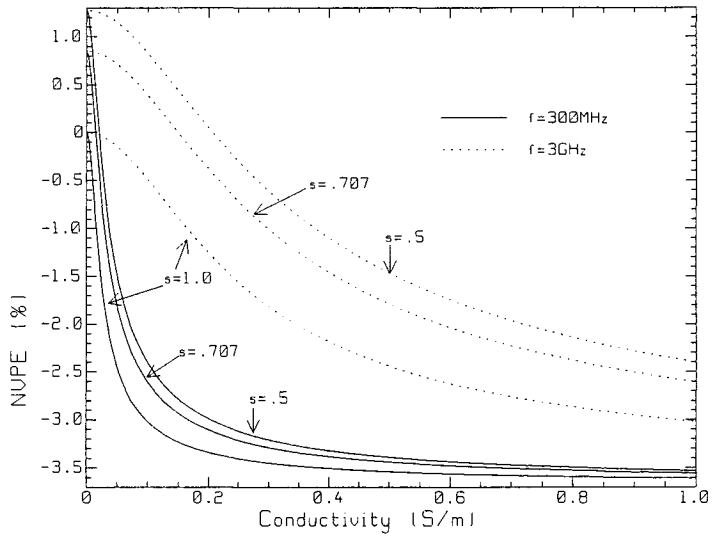

Fig. 9-3 Numerical velocity percentage error vs. conductivity.

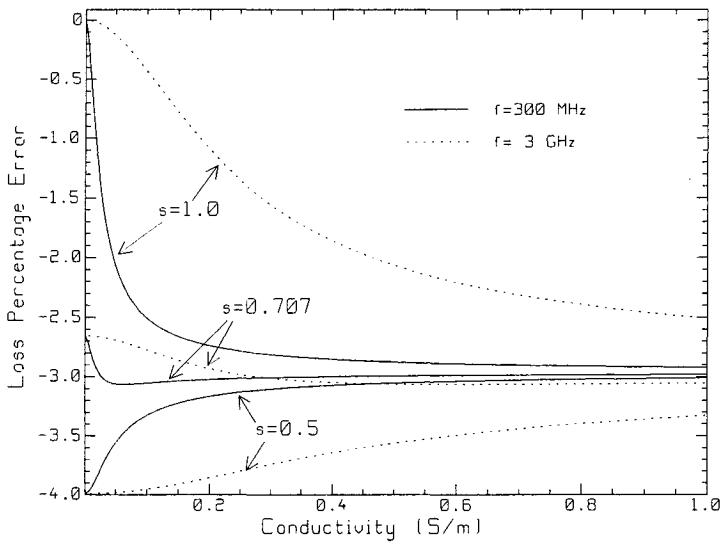

Fig. 9-4 Numerical loss percentage error vs. conductivity.

9-3, the numerical velocity can be larger or smaller than the physical velocity, and for "large" conductivity greater than 0.2 mS/m, the error is always smaller at the higher frequency. In Fig. 9-4, the numerical loss is larger than the physical value at both frequencies. The fact that numerical loss is always larger than its physical value indicates that the conventional FDTD over-estimates the absorption of electromagnetic energy in lossy media, leading to smaller-than-physical field strength in lossy materials.

9.3 Numerical Dispersion and Loss for Unconditionally-Stable Methods in 1D

Since unconditionally-stable methods use a larger time step size than Yee's FDTD, the numerical dispersion and loss will be larger than that of Yee's FDTD method. This section will show that though the numerical dispersion in a lossless material is the same for ADI and CN, it is different in the lossy case. The CN method has higher accuracy than the ADI method.

9.3.1 ADI-FDTD

In the lossy case, Eqn. (9-1) is often used to guide the formulation of the 1D ADI as [85] [104], which is written as

$$E_y^{n+1/2} = a_{0,ADI} E_y^n - a_{1,ADI} D_x H_z^{n+1/2} \tag{9-11a}$$

$$H_z^{n+1/2} = H_z^n - a_2 D_x E_y^{n+1/2} \tag{9-11b}$$

$$E_y^{n+1} = a_{0,ADI} E_y^{n+1/2} - a_{1,ADI} D_x H_z^{n+1/2} \tag{9-12a}$$

$$H_z^{n+1} = H_z^{n+1/2} - a_2 D_x E_y^{n+1/2} \tag{9-12b}$$

where $a_{0,ADI} = (1-\sigma\Delta t/4\varepsilon)/(1+\sigma\Delta t/4\varepsilon)$, $a_{1,ADI} = (\Delta t/2\varepsilon)/(1+\sigma\Delta t/4\varepsilon)$, and $a_2 = \Delta t/2\mu$. Using the matrix form, it is formulated as

$$\begin{pmatrix} 1 & a_{1,ADI} D_x \\ a_2 D_x & 1 \end{pmatrix} W^{n+1/2} = \begin{pmatrix} a_{0,ADI} & 0 \\ 0 & 1 \end{pmatrix} W^n \tag{9-13a}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} W^{n+1} = \begin{pmatrix} a_{0,ADI} & -a_{1,ADI} D_x \\ -a_2 D_x & 1 \end{pmatrix} W^{n+1/2} \tag{9-13b}$$

where $W = \begin{pmatrix} E_y & H_z \end{pmatrix}^T$. The two sub-marching procedures can be combined into one as

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} W^{n+1} = \begin{pmatrix} a_{0,ADI} & -a_{1,ADI} D_x \\ -a_2 D_x & 1 \end{pmatrix} \begin{pmatrix} 1 & a_{1,ADI} D_x \\ a_2 D_x & 1 \end{pmatrix}^{-1} \begin{pmatrix} a_{0,ADI} & 0 \\ 0 & 1 \end{pmatrix} W^n \tag{9-14}$$

The amplification polynomial and the amplification factor ξ can be obtained as

$$\left(1-r_{ADI}^2\right)\xi^2 - \left(1+a_{0,ADI}^2+(1+a_{0,ADI})r_{ADI}^2\right)\xi + a_{0,ADI}\left(a_{0,ADI}-r_{ADI}^2\right) \tag{9-15a}$$

$$\xi = \frac{\left(1 + a_{0,ADI}^2 + (1 + a_{0,ADI}) r_{ADI}^2\right) \pm \sqrt{\left(1 + a_{0,ADI}^2 + (1 + a_{0,ADI}) r_{ADI}^2\right)^2 - 4a_{0,ADI}\left(1 - r_{ADI}^2\right)\left(a_{0,ADI} - r_{ADI}^2\right)}}{\left(1 - r_{ADI}^2\right)} \tag{9-15b}$$

where $r_{ADI} = 2a_{1,ADI}a_2 \sinh(\gamma\Delta x/2)/\Delta x$. Then substituting $\xi = \exp(J\omega\Delta t)$ into Eqn. (9-15a) and after some manipulations obtains the numerical dispersion-loss equations

$$\begin{aligned}(1 + a_{0,ADI}^2)\tan^2(\omega\Delta t/2) = {} & 2a_{1,ADI}a_2(1 + a_{0,ADI})[1 - \cosh(\alpha\Delta x)\cos(\beta\Delta x)]/\Delta x^2 \\ & + 2a_{1,ADI}a_2(1 - a_{0,ADI})\tan(\omega\Delta t/2)\sinh(\alpha\Delta x)\sin(\beta\Delta x)]/\Delta x^2\end{aligned} \tag{9-16a}$$

$$\begin{aligned}(1 - a_{0,ADI}^2)\tan(\omega\Delta t/2) = {} & 2a_{1,ADI}a_2(1 + a_{0,ADI})\sinh(\alpha\Delta x)\sin(\beta\Delta x)/\Delta x^2 \\ & - 2(1 - a_{0,ADI})a_{1,ADI}a_2\tan(\omega\Delta t/2)(1 - \cosh(\alpha\Delta x)\cos(\beta\Delta x))/\Delta x^2\end{aligned} \tag{9-16b}$$

The relation is more complicated than in Yee's FDTD. The numerical dispersion and numerical loss from Eqn. (9-16) and numerical experiments are shown in Fig. 9-5 and Fig. 9-6, for mesh density from 40 CPW to 100 CPW and Courant number from 1 to 4. It can be seen that the numerical dispersion and loss errors decrease as the mesh density increases, and increases as the Courant number or the time step size increases. This differs from the behavior of the explicit, conditionally-stable methods.

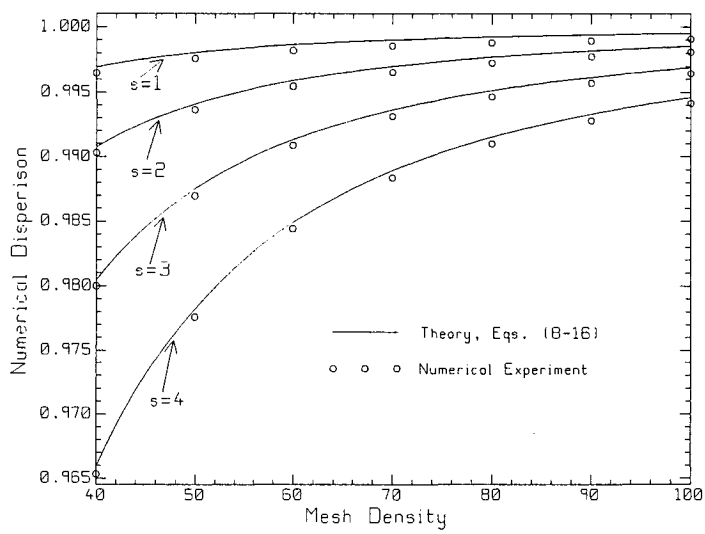

Fig. 9-5 Numerical dispersion of ADI with σ = 0.001 S/m.

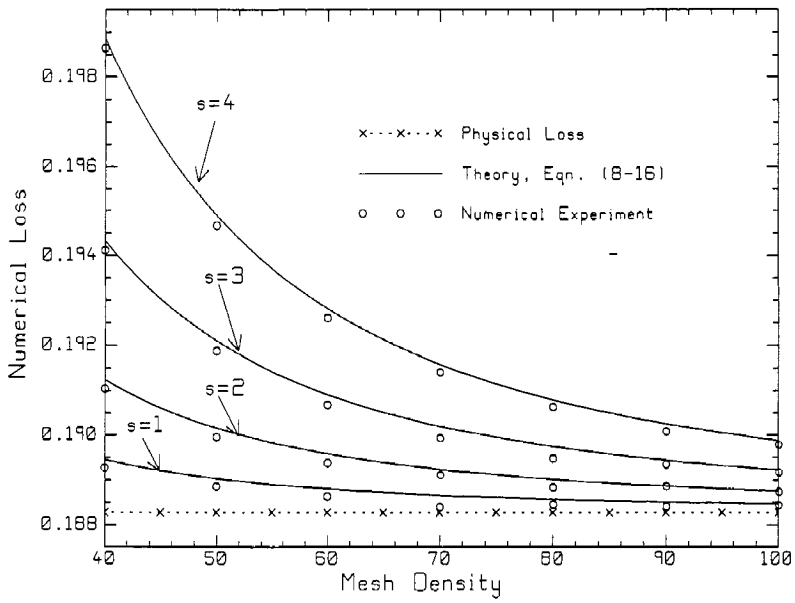

Fig. 9-6 Numerical loss of ADI with σ = 0.001 S/m.

9.3.2 CN-FDTD

From the lossy Maxwell's Equations, the following CN scheme can be formulated

$$E_y^{n+1} = a_{0,CN} E_y^n - a_{1,CN} D_x \left(H_z^{n+1} + H_z^n \right) \tag{9-17a}$$

$$H_z^{n+1} = H_z^n - a_2 D_x \left(E_y^{n+1} + E_y^n \right) \tag{9-17b}$$

where $a_{0,CN} = (1 - \sigma\Delta t / 2\varepsilon)/(1 + \sigma\Delta t / 2\varepsilon)$, $a_{1,CN} = (\Delta t / 2\varepsilon)/(1 + \sigma\Delta t / 2\varepsilon)$. For simplicity, defining the following matrices

$$D_1 = \begin{pmatrix} 0 & a_{1,CN} D_x \\ a_2 D_x & 0 \end{pmatrix} \qquad D_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{9-18}$$

$$I_{CN} = \begin{pmatrix} a_{0,CN} & 0 \\ 0 & 1 \end{pmatrix} \tag{9-19}$$

then the CN scheme can be written in matrix form as

$$(I - D_1 - D_2) W^{n+1} = (I_{CN} + D_1 + D_2) W^n \tag{9-20}$$

where I_2 is an 2×2 identity matrix in the 1D case. Then Eqn. (9-20) can be factorized. Using the efficient-splitting scheme introduced in Chapter 5, the lossy CN scheme can be split into two sub-steps as

$$(I-D_1)W^* = (I_{CN}+D_1+2D_2)W^n \tag{9-21a}$$

$$(I-D_2)W^{n+1} = W^* - D_2 W^n \tag{9-21b}$$

The actual update equations are

$$(1-a_{1,CN}a_2D_{2x})E_y^{n+1} = a_{0,CN}E_y^n - 2a_{1,CN}D_xH_z^n + a_{1,CN}a_2D_{2x}E_y^n \tag{9-22a}$$

$$H_z^{n+1} = H_z^n - a_2D_x\left(E_y^{n+1}+E_y^n\right) \tag{9-22b}$$

and Eqn. (9-17b) if solving implicitly for the electric field, or

$$(1-a_{1,CN}a_2D_{2x})H_z^{n+1} = H_z^n - a_2(1+a_{0,CN})D_xE_y^n + a_{1,CN}a_2D_{2x}H_z^n \tag{9-23a}$$

$$E_y^{n+1} = a_{0,CN}E_y^n - a_{1,CN}D_x\left(H_z^{n+1}+H_z^n\right) \tag{9-23b}$$

and Eqn. (9-17a) if solving implicitly for the magnetic field. The amplification polynomial and the amplification factor ξ can be obtained as

$$\left(1-r_{CN}^2\right)\xi^2 - \left(1+a_0+2r_{CN}^2\right)\xi + \left(a_0-r_{CN}^2\right) \tag{9-24a}$$

$$\xi = \frac{\left(1+a_0+2r_{CN}^2\right) \pm \sqrt{\left(1+a_0+2r_{CN}^2\right)^2 - 4\left(1-r_{CN}^2\right)\left(a_0-r_{CN}^2\right)}}{\left(1-r_{CN}^2\right)} \tag{9-24b}$$

where $r_{CN} = 2a_{1,CN}a_2\sinh(\gamma\Delta x/2)/\Delta x$. Then substituting $\xi = \exp(J\omega\Delta t)$ into Eqn. (9-24a), and after some manipulations obtains

$$(1+a_{0,CN})\tan^2(\omega\Delta t/2) = 4a_{1,CN}a_2[1-\cosh(\alpha\Delta x)\cos(\beta\Delta x)]/\Delta x^2 \tag{9-25a}$$

$$(1-a_{0,CN})\tan(\omega\Delta t/2) = 4a_{1,CN}a_2\sinh(\alpha\Delta x)\sin(\beta\Delta x)/\Delta x^2 \tag{9-25b}$$

Compared to Eqs. (9-5) and (9-16), CN has similar numerical dispersion and loss relation to Yee's FDTD, and the relation is simpler than that for ADI-FDTD. The numerical dispersion predicated from Eqn. (9-25) and that from numerical experiments are shown in Fig. 9-7 with good agreement, for mesh density from 40 CPW to 100 CPW and Courant number from 1 to 4 at $\sigma = 1\text{mS/m}$. At such small value of conductivity, the numerical dispersion of CN is about the same as ADI. However, the numerical loss error is much smaller than ADI. The values of the numerical loss at Courant number 1 to 4 are very

close, and the differences among them are smaller than the accuracy of the computational method used to extract the value of numerical loss, thus will not be graphed. Table 9-1 gives the numerical loss data for the CN scheme. Note that to show the small difference, both the theoretical data and the experimental data have not been rounded.

Table 9-1 Numerical loss in 1D CN-FDTD.

Mesh density >>	40	48	64	72	80	96
s=1 Eqn. (9-25)	0.188867	0.188686	0.188509	0.188460	0.188427	0.188382
s=1 Experiment	0.188678	0.188443	0.188355	0.188353	0.188267	0 .188063
s=2 Eqn. (9-25)	0.188875	0.188690	0.188511	0.188461	0.188428	0.188383
s=2 Experiment	0.188671	0.188494	0.188250	0.188240	0.188320	0.188314
s=3 Eqn. (9-25)	0.188890	0.188698	0.188513	0.188462	0.188429	0.188383
s=3 Experiment	0.188677	0.188495	0.188325	0.188304	0.188154	0.188309
s=4 Eqn. (9-25)	0.188911	0.188708	0.188517	0.188465	0.188431	0.188383
s=4 Experiment	0.188700	0.188500	0.188301	0.188238	0.188268	0.188116

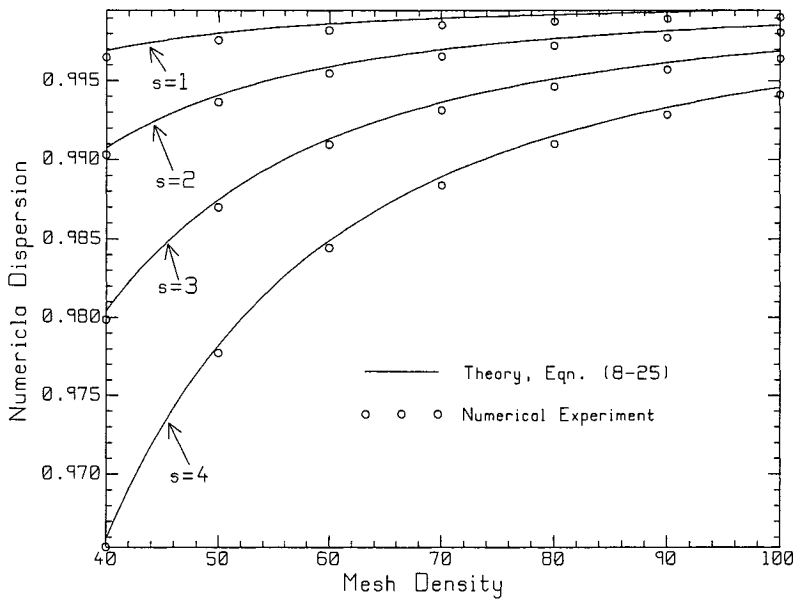

Fig. 9-7 Numerical dispersion of CN with $\sigma = 0.001$ S/m.

9.3.3 Discussion

A. Factorization errors

The factorization error for CN can be obtained from Eqn. (9-20) and (9-21) as

$$\Delta D_{CN} = D_1 D_2 \left(W^{n+1} - W^n\right) \tag{9-26}$$

In the 1D case, because D_2 is a zero matrix, there is no factorization error for CN.

For ADI, since the formulation of the coefficient a_0 is different from CN, it cannot be written as a perturbation of the CN scheme. Thus only the factorization error is discussed. Define

$$D_1 = \begin{pmatrix} 0 & a_{1,ADI} D_x \\ a_2 D_x & 0 \end{pmatrix} \qquad D_2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{9-27}$$

$$I_{ADI} = \begin{pmatrix} a_{0,ADI} & 0 \\ 0 & 1 \end{pmatrix} \tag{9-28}$$

then the ADI-FDTD can be factorized and split as

$$(I - D_1) W^{n+1/2} = (I_{ADI} + D_2) W^n \tag{9-29a}$$

$$(I - D_2) W^{n+1} = (I_{ADI} + D_1) W^{n+1/2} \tag{9-29b}$$

which is a perturbation to the following equation

$$(I - D_1 - D_2) W^{n+1} = (I_{ADI} + D_1 + D_2) W^n \tag{9-30}$$

Thus the factorization error for ADI-FDTD is

$$(I_{ADI}^2 - I) W^n + (I_{ADI} - I)(D_1 + D_2) W^n + D_1 D_2 (W^{n+1} - W^n) \tag{9-31}$$

It can be seen that in lossy case, the error is 1st order in time. Compared to Eqn. (9-26), ADI has larger factorization error than CN.

B. Intrinsic temporal numerical dispersion and loss

As discussed previously [153] [154], unconditionally-stable methods have intrinsic temporal numerical error due to the relaxation of the Courant constraint. When the mesh size goes to zero, the intrinsic numerical phase constant and loss for ADI and CN can be obtained with some manipulation as

$$\beta_t = \sqrt{\frac{A}{2}}\left(\sqrt{1+\frac{B^2}{A^2}}+1\right)^{1/2} \qquad \alpha_t = \sqrt{\frac{A}{2}}\left(\sqrt{1+\frac{B^2}{A^2}}-1\right)^{1/2} \tag{9-32}$$

Where

$$A = \begin{cases} \dfrac{\tan^2(\omega\Delta t/2)}{(c\Delta t/2)^2}\dfrac{(1-q)^2}{1-q^2\tan^2(\omega\Delta t/2)} & ADI \\ \dfrac{\tan^2(\omega\Delta t/2)}{(c\Delta t/2)^2} & CN \end{cases} \tag{9-33a}$$

$$B = \begin{cases} \sigma\mu\dfrac{\tan(\omega\Delta t/2)}{\Delta t/2}\dfrac{1-(1+q^2)\tan^2(\omega\Delta t/2)/2}{1-q^2\tan^2(\omega\Delta t/2)} & ADI \\ \sigma\mu\dfrac{\tan(\omega\Delta t/2)}{\Delta t/2} & CN \end{cases} \tag{9-33b}$$

$$q = \frac{\sigma\Delta t}{4\varepsilon} \tag{9-34}$$

where subscript t stands for intrinsic temporal. It can be seen that when σ goes to zero, the intrinsic loss constant is zero, and the numerical phase constant reaches its intrinsic temporal value [154]. As the time step size increases with non-zero conductivity, the deviation to their physical values increases, since $\tan(y)/y$ increases greatly. Fig. 9-8 and Fig. 9-9 show the intrinsic temporal numerical velocity and numerical loss percent error (NVPE and NLPE) at time step size relative to the Nyquist limit. With conductivity increases, the error for numerical loss decreases and the error for velocity increases for CN. For ADI, the relation of NLPE and NVPE with time step size is complicated, but generally they are much larger than those for the CN scheme.

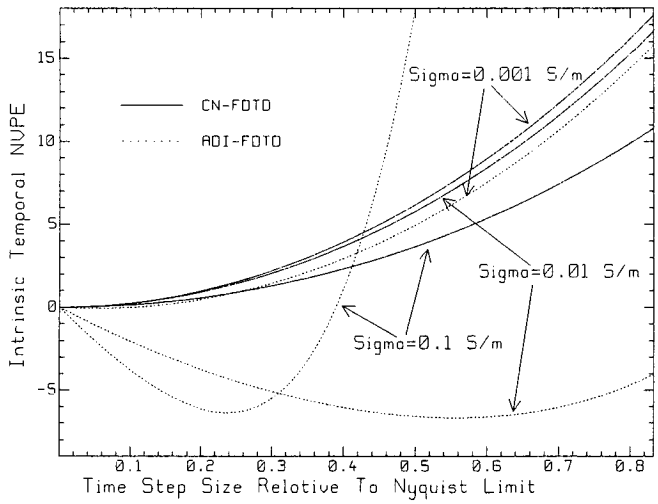

Fig. 9-8 Intrinsic temporal numerical velocity percent error for CN and ADI.

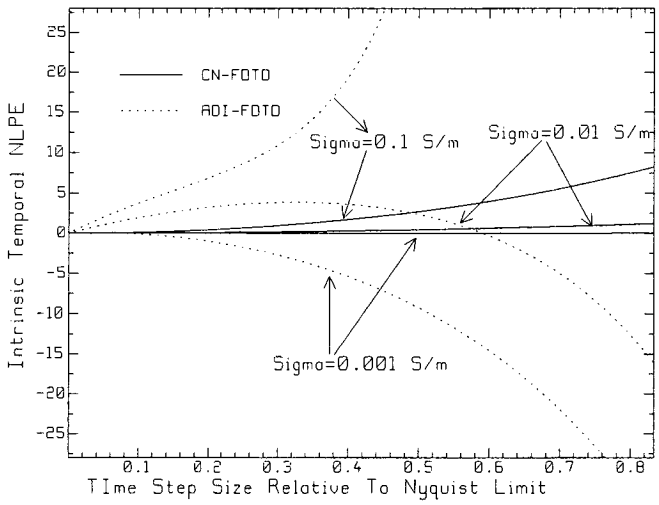

Fig. 9-9 Intrinsic temporal numerical loss percent error for CN and ADI.

C. Intrinsic spatial dispersion and loss

When the time step size goes to zero, the numerical error still exists due to spatial discretization. However, since $\Delta t \to 0$ makes the term $\sigma \Delta t \to 0$, the numerical error for CN and ADI becomes the same. Such error is termed as "intrinsic spatial error" because in the lossy case it depends on the approximation of the spatial derivatives and the

approximation of σE_y, which uses the time averaged approximation in this Chapter. The relation between the intrinsic-spatial numerical phase constant and loss is simplified as

$$\omega^2 \Delta x^2 / 2c^2 = 1 - \cosh(\alpha \Delta x)\cos(\beta \Delta x) \quad (9\text{-}35a)$$

$$\sigma \mu \omega \Delta x^2 / 2 = \sinh(\alpha \Delta x)\sin(\beta \Delta x) \quad (9\text{-}35b)$$

It can be seen that even when there is no time discretization, the numerical phase constant and loss constant still do not agree with their physical counterparts.

D. Numerical dispersion and loss with time step size

Fig. 9-10 shows the numerical dispersion percent error as a function of time step size (Courant number), at a mesh density of 50 CPW. Generally, numerical dispersion error increases as the time step size increases. For small conductivity, such as $\sigma = 0.001 S/m$, the numerical dispersion is about the same for both CN and ADI. However, the numerical loss percent error for ADI is much larger than CN, and becomes larger as conductivity is small because it is expressed in a percentage. For clarity, Fig. 9-11 shows the numerical-loss absolute error on a logarithmic scale. It can be seen that, as the conductivity increases, the deviation becomes larger.

E. Numerical dispersion and loss with conductivity

Fig. 9-12 and Fig. 9-13 graph the numerical dispersion and loss percent error as a function of conductivity at mesh density of 50 CPW. Since it is in percentage, the errors decrease as conductivity increases. However, the absolute error increases as conductivity increases.

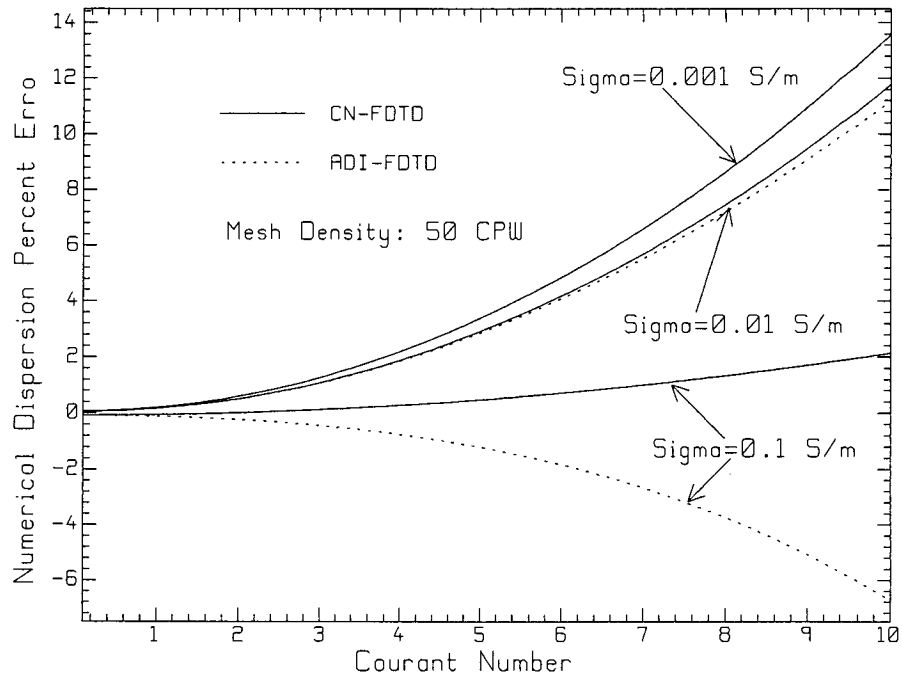

Fig. 9-10 Numerical dispersion percent error versus Courant number at a mesh density of 50 CPW.

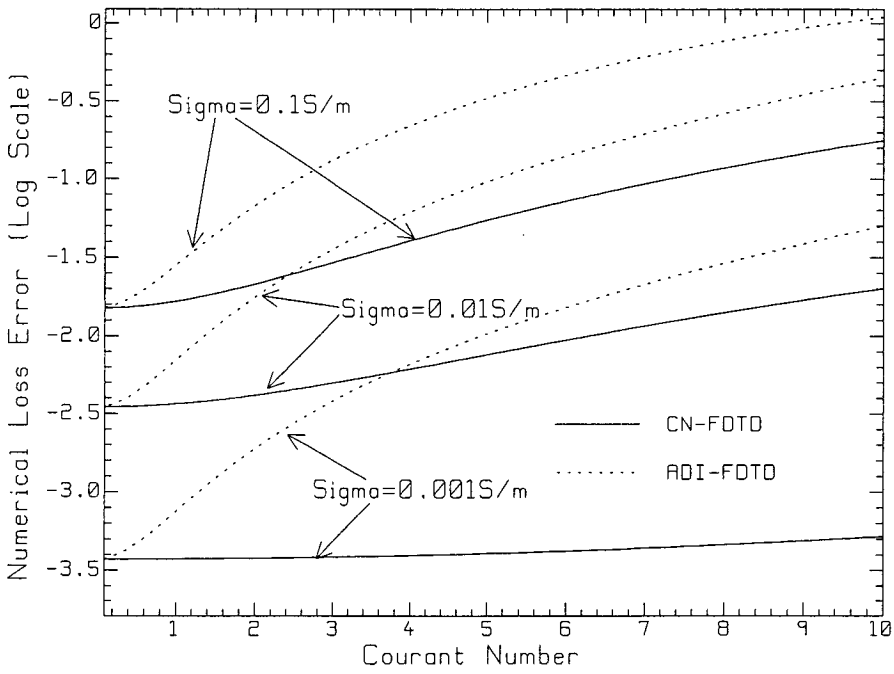

Fig. 9-11 Numerical loss percent error versus Courant number at a mesh density of 50 CPW.

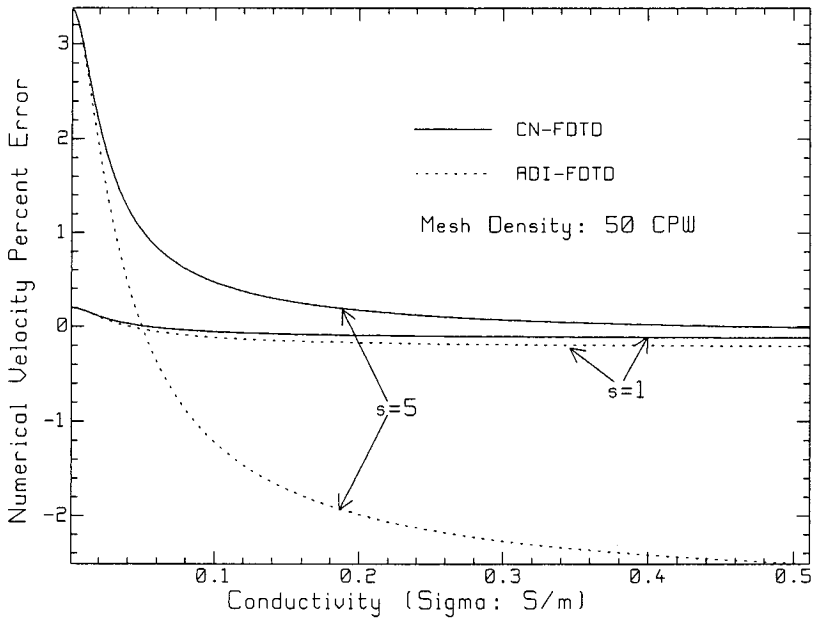

Fig. 9-12 Numerical dispersion percent error versus conductivity at mesh density 50 CPW.

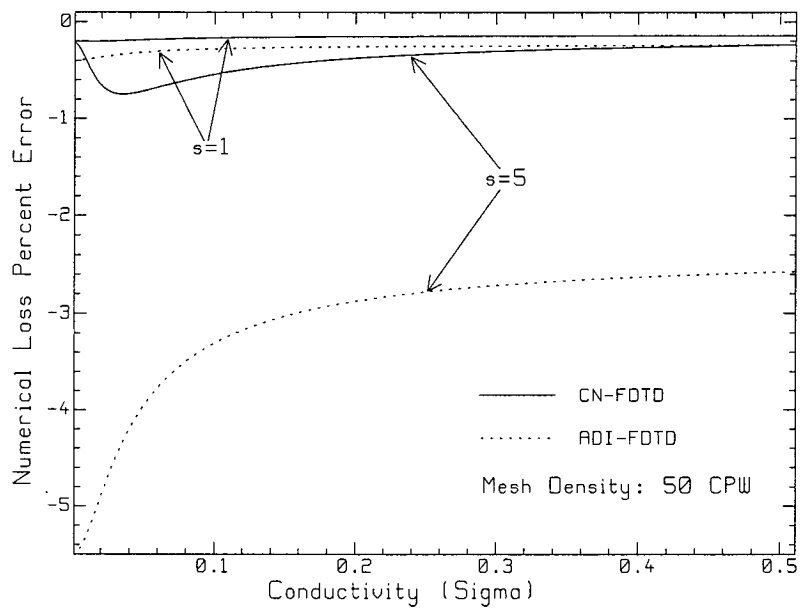

Fig. 9-13 Numerical loss percent error versus conductivity at mesh density 50 CPW.

9.3.4 High-accuracy ADI formulation based on the original CN scheme

From above discussion it is known that ADI is less accurate as CN. This is because of the intuitive treatment of the loss term. From Maxwell's Equations (2-6a) and (2-6b), the CN scheme can be generally written (for 1D, 2D and 3D) as

$$w^{n+1} = w^n - (D_1 + D_2 + D_L)(w^{n+1} + w^n) \tag{9-36a}$$

$$D_1 + D_2 = -\frac{\Delta t}{2}\begin{pmatrix} 0 & \varepsilon^{-1}\nabla\times \\ \mu^{-1}\nabla\times & 0 \end{pmatrix} \tag{9-36b}$$

$$D_L = -\frac{\Delta t}{2}\begin{pmatrix} Ia_e & 0 \\ 0 & Ia_h \end{pmatrix} \tag{9-36c}$$

where I is an identity matrix with the same dimension as ∇, $a_e = \sigma\varepsilon^{-1}$ and $a_h = \sigma^*\mu^{-1}$. For non-magnetic materials, $a_h \equiv 0$. Compared to Eqn. (4-11), there is an extra matrix D_L. Apparently Eqn. (9-36) can be factorized as

$$(I - D_1 - D_L)(I - D_2)w^{n+1} = (I + D_1 + D_L)(I + D_2)w^n \tag{9-37a}$$

$$(I - D_1)(I - D_2 - D_L)w^{n+1} = (I + D_1)(I + D_2 + D_L)w^n \tag{9-37b}$$

which can be decoupled as Eqn. (4-12) for ADI, and Eqs. (5-2), (6-10) and (6-11) for CNDS. The factorization error in Eqn. (9-37a) is $D_1D_2 + D_LD_2$, which is 2nd-order accurate in time. Thus, with the formulation Eqn. (9-37), ADI has the same numerical accuracy as the CN scheme in the 1D case. Note that the factorization error in Eqn. (9-37b) is $D_1D_2 + D_1D_L$. In 1D case, Eqn. (9-37a) has higher accuracy than Eqn. (9-37b). It has been found that the published papers in ADI-PML usually use the intuitive formulations similar to Eqs. (9-11) and (9-12), and have large reflection errors for large Courant numbers. If the formulation Eqn. (9-37) is used, the performance of the PML may be improved. This issue needs further investigation.

9.4 Numerical Dispersion and Loss in Higher Dimensions

Since the magnitude of the electromagnetic field decreases rapidly with spatial distance in the 2D and 3D lossy media, this Section will only discuss the theoretical numerical dispersion and loss for Yee's FDTD, CN-FDTD and CNCS-FDTD. For ADI and CNDS, their amplification factors are quite complicated and will not be discussed. The analysis will illustrate the behaviors of numerical loss and numerical dispersion with direction of travel for a given mesh density. For conciseness, the update equations, the amplification polynomial and the amplification factor are given without further description. In 2D case, the following parameters are defined: $\gamma_x = \alpha_x + J\beta_x$; $\gamma_y = \alpha_y + J\beta_y$; $\alpha_x = \alpha\cos\phi$; and $\alpha_y = \alpha\sin\phi$.

9.4.1 2D Yee's FDTD

For 2D Yee's FDTD, the update equations are

$$H_z^{n+1/2} = H_z^{n-1/2} + a_2 D_y E_x^n - a_2 D_x E_y^n \tag{9-38a}$$

$$E_x^{n+1} = a_0 E_x^n + a_1 D_y H_z^{n+1/2} \tag{9-38b}$$

$$E_y^{n+1} = a_0 E_y^n - a_1 D_x H_z^{n+1/2} \tag{9-38c}$$

The amplification polynomial is

$$P(\xi) = (a_0 - \xi)\left(\xi^2 + a_0 - (1 + a_0 + r_x^2 + r_y^2)\xi\right) \tag{9-39a}$$

The amplification factor is

$$\xi = \frac{\left(1 + a_0 + r_x^2 + r_y^2\right) \pm \sqrt{\left(1 + a_0 + r_x^2 + r_y^2\right)^2 - 4a_0}}{2} \tag{9-39b}$$

The numerical dispersion and loss relation is derived as

$$\frac{2\sin^2(\omega\Delta t/2)}{(c\Delta t)^2} = \left(\frac{1-\cosh(\alpha_x\Delta x)\cos(\beta_x\Delta x)}{\Delta x^2} + \frac{1-\cosh(\alpha_y\Delta y)\cos(\beta_y\Delta y)}{\Delta y^2}\right) \tag{9-40a}$$

$$\sigma\eta\frac{\sin(\omega\Delta t)}{c\Delta t} = 2\left(\frac{\sinh(\alpha_x\Delta x)\sin(\beta_x\Delta x)}{\Delta x^2} + \frac{\sinh(\alpha_y\Delta y)\sin(\beta_y\Delta y)}{\Delta y^2}\right) \tag{9-40b}$$

9.4.2 2D CN FDTD

For 2D CN FDTD, the difference matrices are constructed as

$$D_1=\begin{pmatrix}0 & 0 & a_2D_y\\ 0 & 0 & 0\\ a_2D_y & 0 & 0\end{pmatrix}\quad D_2=\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & -a_1D_x\\ 0 & -a_2D_x & 0\end{pmatrix}\quad D_L=\begin{pmatrix}a_e & 0 & 0\\ 0 & a_e & 0\\ 0 & 0 & 0\end{pmatrix} \tag{9-41}$$

The amplification polynomial is

$$P(\xi)=\left(a_{0,CN}-\xi\right)\left(\left(1-\left(r_x^2+r_y^2\right)\right)\xi^2+a_{0,CN}-\left(r_x^2+r_y^2\right)-\left(1+a_{0,CN}+2\left(r_x^2+r_y^2\right)\right)\xi\right) \tag{9-42a}$$

The propagating-mode amplification factor is solved as

$$\xi=\frac{\left(1+a_{0,CN}+2r_x^2+2r_y^2\right)\pm\sqrt{\left(1+a_{0,Cn}+2r_x^2+2r_y^2\right)^2-4\left(1-r_x^2-r_y^2\right)\left(a_{0,CN}-r_x^2-r_y^2\right)}}{2\left(1-r_x^2-r_y^2\right)} \tag{9-42b}$$

The numerical dispersion and loss relation is

$$\frac{\left(1+a_{0,CN}\right)\tan^2\left(\omega\Delta t/2\right)}{4a_{1,CN}a_2}=\frac{1-\cosh(\alpha_x\Delta x)\cos(\beta_x\Delta x)}{\Delta x^2}+\frac{1-\cosh(\alpha_y\Delta y)\cos(\beta_y\Delta y)}{\Delta y^2} \tag{9-43a}$$

$$\frac{\left(1-a_{0,CN}\right)\tan(\omega\Delta t/2)}{4a_{1,CN}a_2}=\frac{\sinh(\alpha_x\Delta x)\sin(\beta_x\Delta x)}{\Delta x^2}+\frac{\sinh(\alpha_y\Delta y)\sin(\beta_y\Delta y)}{\Delta y^2} \tag{9-43b}$$

9.4.3 2D CNCS methods

The coupled equations for the electric field components are similar to the lossless case except that $a_{0,CN}$ replaces unity before E_x^n and E_y^n in Eqn. (5-9), as given in Eqn. (9-44). The equations can be decoupled using the methods in Chapter 5. The amplification polynomial and amplification factor are given in Eqn. (9-45). Compared to CN in Section 9.4.2, CNCS has an extra higher order cross term due to factorization. Because of its complexity, the numerical dispersion and loss relation will not be given, but can be obtained from the amplification factor.

The coupled equations for the electric field components are

$$(1-b^2D_{2y})E_x^{n+1}+b^2D_xD_yE_y^{n+1}=(a_{0,CN}+b^2D_{2y})E_x^n-b^2D_xD_yE_y^n+2a_1D_yH_z^n \tag{9-44a}$$

$$(1-b^2D_{2x})E_y^{n+1}+b^2D_xD_yE_x^{n+1}=(a_{0,CN}+b^2D_{2x})E_y^n-b^2D_xD_yE_x^n-2a_1D_xH_z^n \tag{9-44b}$$

The amplification polynomial and the propagating-mode amplification factor are

$$P(\xi)=\left(a_{0,CN}-\xi\right)\left(r_-^4\xi^2+a_{0,CN}-1+r_-^4\right)-\left(1+a_{0,CN}+2r_+^4\right)\xi \tag{9-45a}$$

$$\xi=\frac{\left(a_{0,CN}-1+2r_+^4\right)\pm\sqrt{\left(a_{0,CN}-1+2r_+^4\right)^2-4r_-^4\left(a_{0,CN}-1+r_-^4\right)}}{2r_-^4} \tag{9-45b}$$

where

$$r_-^4=\left(1-a_1a_2r_x^2\right)\left(1-a_1a_2r_y^2\right) \tag{9-46a}$$

$$r_+^4=\left(1+a_1a_2r_x^2\right)\left(1+a_1a_2r_y^2\right) \tag{9-46b}$$

9.4.4. Discussion

From the expressions for the amplification factors of the Yee's and CN methods, it can be seen that the 2D case is just a direct extension of 1D case. Fig. 9-14 and Fig. 9-15 show the numerical dispersion and numerical loss for Yee's FDTD at mesh density 10 CPW at the Courant limit, and Fig. 9-16 and Fig. 9-17 for CN-FDTD at mesh density 50 CPW at Courant number 1 and 5, with conductivity of 0.001 S/m, and 0.005S/m. It can be seen that the numerical dispersion and numerical loss have higher accuracy along the diagonals than along the axes for both methods; and both the numerical dispersion and numerical loss are anisotropic. However, the anisotropy for CN-FDTD is smaller than for Yee's FDTD, and the anisotropy of the numerical dispersion is much smaller than that of the numerical loss for CN-FDTD. In addition, the numerical dispersion and numerical loss error increase for Yee's FDTD as the time step size decreases; in contrast, the numerical dispersion and loss error increase for CN-FDTD with time step size increases. The numerical error for CNCS is similar to CN-FDTD.

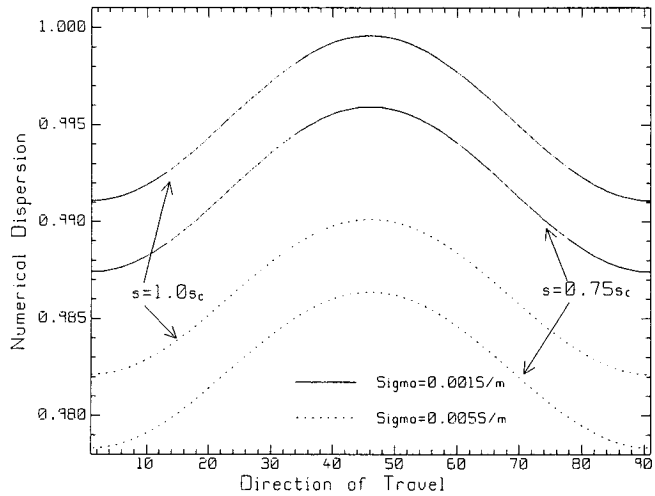

Fig. 9-14 Numerical dispersion of Yee's FDTD at mesh density 10 CPW.

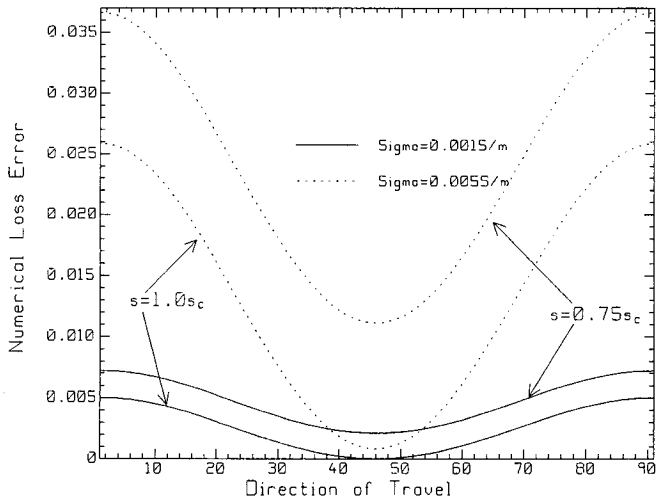

Fig. 9-15 Numerical loss error of Yee's FDTD at mesh density 10 CPW.

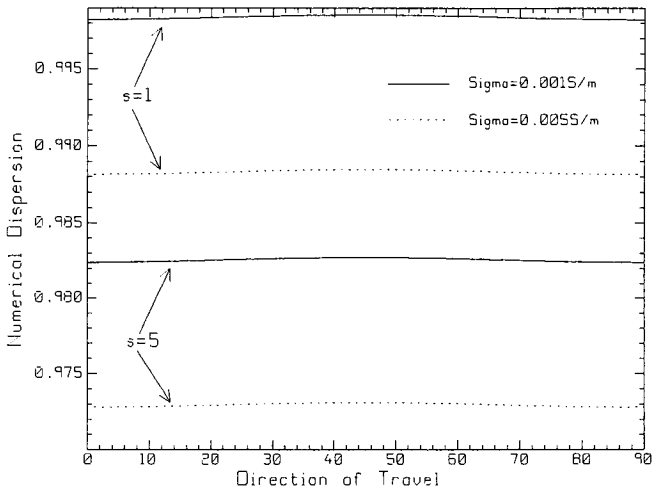

Fig. 9-16 Numerical dispersion of CN-FDTD at mesh density 50 CPW.

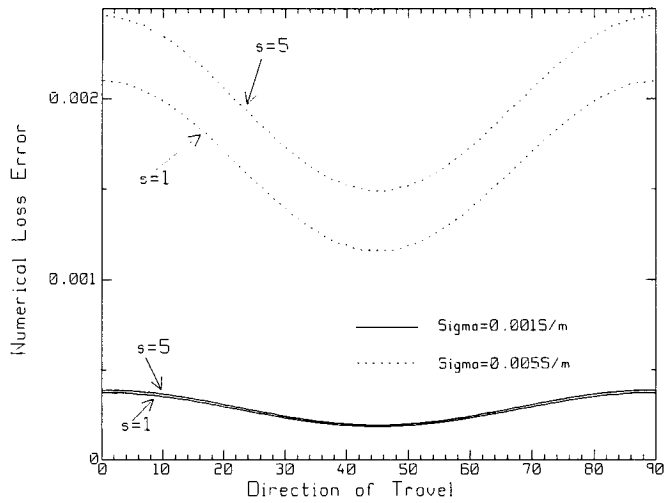

Fig. 9-17 Numerical loss error of CN-FDTD at mesh density 50 CPW.

9.4.5 3D Yee and CN FDTD

For the 3D case, the numerical dispersion and loss relations for Yee's FDTD and CN FDTD can be derived similarly to Eqn. (9-40) and Eqn. (9-43) except that the term in *z*-axis is added. The equations will not be given.

9.4.6 Anisotropies

Generally it is difficult to compute the numerical dispersion and loss without using a root-finding algorithm because of the direction-of-travel dependence. However, for the direction of travel along the diagonals of a square or cubic mesh, the numerical dispersion term and the loss term can be separated as follows [164]

$$\sinh^2(\alpha\Delta x/\sqrt{d}) = 2r_t^2\left(r_t^2 - 1 + r_t^2\tan^2\delta \pm \sqrt{(1+\tan^2\delta)(r_t^4\tan^2\delta + (1-r_t^2)^2)}\right) \quad (9\text{-}47a)$$

$$\sin^2(\beta\Delta x/\sqrt{d}) = 2r_t^2\left(1 - r_t^2(1+\tan^2\delta) \pm \sqrt{(1+\tan^2\delta)(r_t^4\tan^2\delta + (r_t^2-1)^2)}\right) \quad (9\text{-}47b)$$

where $r_t = \sin(\omega\Delta t/2)/\left(s\sqrt{d}\right)$ for Yee's FDTD, and $r_t = \tan(\omega\Delta t/2)/\left(s\sqrt{d}\right)$; and $d = 3$ for 3D, 2 for 2D, 1 for 1D and along axes; the physical loss tangent is $\tan\delta_c = \sigma/\omega\varepsilon$, and the term $\tan\delta = \left((\omega\Delta t/2)/\tan(\omega\Delta t/2)\right)\tan\delta c$ may be called "numerical loss tangent".

Using Eqn. (47) the anisotropies for the numerical dispersion and loss can be easily calculated. Define the anisotropy of numerical loss as

$$A_l = \frac{\alpha_a - \alpha_d}{\min\{\alpha_d, \alpha_a\}} \times 100\% \tag{9-48}$$

Fig. 9-18 shows the anisotropies of numerical dispersion A_u defined in Eqn. (4-22) and of numerical loss for Yee's FDTD in the 2D case. 3D anisotropies are similar to Fig. 9-18 but the magnitude is larger. Some data are shown in Table 9-2 for the comparison.

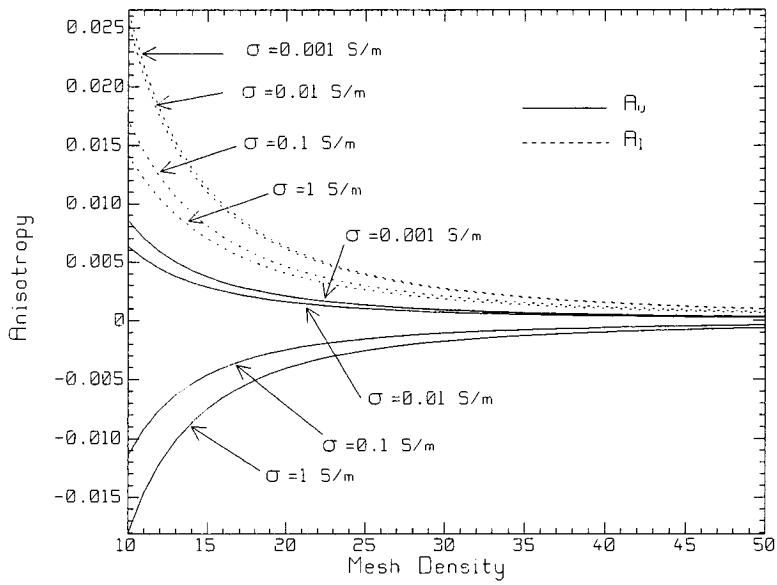

Fig. 9-18 Anisotropies for 2D Yee's FDTD.

Table 9-2 Comparison of the anisotropies in 2D and 3D.

σ (S/m)>>	0.001	0.01	0.1	1
A_u 1n 2D	0.0086	0.0064	-0.0115	-0.0181
A_l in 2D	0.0265	0.0254	0.0171	0.0142
A_u 1n 3D	0.0114	0.0085	-0.0151	-0.0239
A_l in 3D	0.0354	0.0340	0.0232	0.0194

Although the numerical dispersion and loss depend on the time step size (the Courant number), as shown in Fig. 9-14 and Fig. 9-15, the anisotropies are insensitive to the time step size. Thus Fig. 9-18 shows only the anisotropies at the Courant limit for a squared

mesh, at frequency 300 MHz. The conductivities are 0.001, 0.01, 0.1 and 1 S/m, respectively. It can be seen that the anisotropy A_l is always larger than zero, which implies that the numerical loss along the diagonals is always smaller than that along the axes, and more close to its physical value. However, the anisotropy A_u can be larger (σ = 0.001 and 0.01 S/m) or smaller (σ =0.1 and 1 S/m) than zero, which indicates that the numerical velocity along the diagonal can be larger or smaller than that along the axes. Further investigation shows that, when the numerical velocity anisotropy is positive, the numerical velocities for all directions of travel are smaller than the physical velocity; when the numerical velocity anisotropy is negative, all the numerical velocities are larger than the physical value, which has not been found in lossless media. This fact also demonstrates that the numerical velocity along the diagonals has higher accuracy.

In addition, between the positive and the negative anisotropies of the numerical dispersion, there is a zero-anisotropy point. For example, at the conductivity 0.02614 S/m in 2D, at mesh density 10 CPW, frequency 300 MHz, the numerical velocity anisotropy is about 3×10^{-7}, which is almost zero. In the 3D case, the anisotropy is 2×10^{-8} with conductivity 0.02632 S/m. In the lossless case, zero anisotropy means the numerical wave has the same velocity in all the directions of travel. However, further investigation shows that it is not true for the lossy case. For a square mesh with a mesh density of 10 CPW at frequency 300 MHz, at the directions of 22.5° and 67.5°, the numerical dispersion is about 1.00774. At the direction of 0°, 45° and 90°, the numerical dispersion is about 1.00785. The non-uniformity of the velocity is about 10^{-4}, which is highly isotropic. And at this special condition, it can be seen that the directions of 22.5° and 67.5° have higher accuracy of numerical velocity than along axes and the diagonals.

9.5 Conclusion

This Chapter has discussed some important behaviors of FDTD methods in lossy media. In addition to numerical dispersion, the loss constant of the FDTD methods is also different from its physical value, which is reasonable since the phase constant and the loss constant are in the same augment of the Fourier mode. The relations of numerical dispersion and loss are derived analytically for several FETID methods. Numerical experiments were performed in 1D, which agree well with analysis. Similar to numerical dispersion, numerical loss is a function of mesh density, time step size and the conductivity. Coarser cell sizes or larger conductivity causes larger numerical loss error. As the time step decreases, the loss error for Yee's FDTD increases, but the loss error for CN and ADI decreases. The intuitive formulation of ADI, that is, discretizing Maxwell's Equations at each individual sub-step with the ADI methodology in lossless case, has been shown to be a first-order method in time, and it has larger numerical loss error than CN. To have the same accuracy as CN, ADI must be written as a perturbation of the original CN scheme. In the 2D case, the numerical dispersion and loss are anisotropic for Yee's FETID, CN-FETID and CNCS. The numerical dispersion and numerical loss are separated for the diagonal direction of travel for a square mesh or a cubic mesh and along the axes, which can be used to compute the anisotropies. Similar to the lossless case, the numerical wave traveling along the diagonal usually has usually a higher accuracy for both the velocity and the loss constant than along other directions. The numerical velocity for Yee's FDTD can be larger than its physical velocity, which is different from the lossless case. FDTD methods over-estimate the electromagnetic absorption, and thus lead to a smaller-than-physical field strength in lossy materials.

Chapter 10 Summary, Conclusion and Future Work

This thesis has proposed several new FDTD methods with better performances over other methods: smaller dispersion error; less anisotropy; unconditional stability. While working on developing these methods, it has been observed and briefly discussed that there are some new properties, behaviors and mechanisms of FDTD which have not been reported previously or have been neglected. This Chapter first gives such observations, then summarizes the contributions, and finally points out future work to be done.

10.1 Some Observations

It is believed that the advantages of an FDTD method can be fully utilized for certain applications, based on a good understanding of its properties and limitations. The following are the observations which are helpful.

A. Nyquist Criterion, attenuation and time-step-size limits

The Nyquist criterion is very important for all discrete systems. FDTD methods are discrete systems in both time and space; thus the criterion must be obeyed in analyzing some properties of FDTD methods, such as the numerical stability. Section 2.4 has pointed out that, for explicit FDTD methods, the mesh density must be smaller than and not equal to the Nyquist limit. It has been shown in [153-154] and Chapter 7 that though the unconditionally-stable FDTD methods are "numerically" stable for any time step size, there is numerical attenuation that prohibits the use of large time step size over the limit given in Section 7.5, which is smaller than the Nyquist limit. The FDTD community does not use such a large time step size, because of accuracy requirements. However, solving elliptic equations may use a very large time step size where the goal of the time stepping method is to reach steady state as fast as possible, which may lead to an

erroneous result. In addition, if the mesh density along each axis is larger than 2 CPW of the spatial Nyquist limit, the faster-than-light phenomenon cannot occur.

B. Amplification polynomials for stable FDTD methods

From Chapters 2, 3, 4, 5, 6 and 9, the amplification polynomial for a stable FDTD method can be generalized as Eqn. (7-24). In the lossy case of Chapter 9, $\xi - 1$ is replaced by $\xi - a_0$. The stationary solution corresponds to the longitudinal electromagnetic field, and the non-stationary solution corresponds to the transverse electromagnetic field or the propagating mode. The total number of the solutions is the same as the total number of the electromagnetic field components in the Cartesian coordinate system. There should be only two conjugate, distinct solutions for the propagation modes regardless of the dimensions. Otherwise the method will generate spurious solutions.

C. Isotropic error and intrinsic errors

The spatial dispersion error is anisotropic, which is the sole cause of anisotropy for Yee's FDTD Eqn. (2-12) and the CN scheme Eqn. (4-5). When the mesh in very fine, Yee's FDTD tends to be isotropic; and can have no or very little numerical dispersion error due to the Courant constraint.

For ADI and other CN-based methods, because of factorization, some cross terms are implicitly incorporated into the update equations. Thus unlike Yee's FDTD, these unconditionally-stable methods have intrinsic temporal numerical dispersion, and such dispersion is anisotropic. But it is isotropic for the CN scheme only. The CNCS, CNDG and CNAFS methods [151] [156-157] in Chapter 5 and Chapter 6 have much smaller anisotropy than ADI, and can have zero anisotropy at certain mesh densities and Courant

numbers. ADI and CNDS are more anisotropic. Note that the anisotropy depends on the formulation of the spatial derivative approximation and the time step size used. The intrinsic error gives the accuracy limit for a given time step size. The unconditionally-stable methods cannot provide better accuracy than its intrinsic dispersion for whatever fine the mesh is.

D. Splitting schemes

Although the ADI and CNDS methods are based on the same formulation, and they have the same numerical dispersion relation, they have different computational efficiency and discretization error. This arises because the splitting method is different given in Eqs. (4-12) and (5-2). Therefore developing an efficient, highly-accurate splitting scheme is important. In addition, to avoid numerical attenuation, one cannot drop out any cross terms in order to reduce the computational stencil, as pointed out in Section 5.1.

E. Intermediate time step

ADI has an intermediate time step explicitly associated with time step $n+1/2$; and the intermediate field values may be used erroneously to couple to a hybrid scheme. CN-based methods have an intermediate time step that is not specifically associated with any time step [151] [156-157]; thus the erroneous use of the intermediate field values is avoided by this construction.

The intermediate time step $n+1/2$ for ADI causes one sub-step marching procedure to have growth, and the other sub-step marching procedure to have dissipation. However, the growth is exactly cancelled by the dissipation; thus the overall ADI scheme is unconditionally stable [152] [154].

In addition, since ADI uses the specific time value as the intermediate time step, the enforcement of the current or voltage source values must be made at both sub-steps. However, for CN-based methods, the enforcement of such source values is only in the first sub-step; thus it is more effective.

F. Coefficient-modification technique

Though there are several different methods to reduce numerical dispersion, many of them can be generalized into the coefficient-modification techniques as described in Section 3.3. They all modify the coefficients in the update equations and thus modify the speed. Though they all reduce the numerical dispersion error, from Chapter 9 it can be seen that the discretization errors may not be improved as much as the dispersion, if not all of the coefficients are modified by the same amount for a square or cubic mesh [159].

10.2 Summary and Conclusion

The main work of this thesis is summarized and concluded in this section.

A. High-accuracy explicit methods

The high-accuracy explicit methods proposed in Chapter 3 can be generally termed "optimization methods". Instead of pursuing nominal "higher-order" in terms of Taylor Series, the 24-stencil method (O24) [159], neighborhood-average method (NA) and the neighborhood-average-24-stencil method (NA24) are formulated to optimize the exact numerical dispersion relation, not just an approximation of the relation. In a different sense, the methods can have better numerical dispersion within a given sector, such as the axes-optimized method (AOM), the diagonal-optimized method (DOM), and isotropic method (IOM). Actually the numerical dispersion can be optimized within any given sector by the search method given in Section 3.2.4. Chapter 3 gives some analytical

weight formulas in specific cases for illustration. The simple search method is universal and more *powerful* in finding such optimal weight parameters once the accuracy criterion is specified. The significant advantage of the optimized methods presented in this thesis is that the boundary conditions at the material interfaces can be treated efficiently using the current 2^{nd}-order methods, without giving rise to new difficulties.

A general dispersion relation for an explicit FDTD scheme with 2^{nd}-order accuracy in time is given by Eqn. (9-10) in the lossless case. It shows that for explicit methods, there is no cross term, thus explicit methods are more isotropic than many implicit methods. The numerical dispersion depends on the transfer functions of the difference operators. To reduce the numerical error, the transfer functions in the plane wave matrices given in Chapter 9 must be close to unity with direction of travel, mesh density and the Courant number. Thus designing the difference operator is similar to devising a digital filter.

It is also worth mentioning that the time-step-size limit for the optimized methods is a "shunt" combination of the two limits of the individual methods.

B. Numerical dispersion relation for ADI-FDTD

The author was the first to identify that there is inconsistency among the published numerical dispersion relations for ADI-FDTD [160]. This inconsistency has been removed [152] [154] in Chapter 4. Beginning with the analysis from the 1D case, the correct numerical dispersion relation has been obtained using the Fourier analysis for the *individual* sub-marching procedures, instead of the overall one-step analysis [99] [101]. The by-product is the finding of the numerical growth and dissipation in its individual sub-steps, and thus the mechanism of the unconditional stability. In addition, the numerical dispersion relation is numerically validated, and the numerical results extracted

by the matching method [158] described in Chapter 2 agree very well with the theoretical prediction. To accurately extract the numerical phase velocity for the verification of the dispersion relations, the numerical experiments must be carefully designed.

To reduce the numerical dispersion error, an isotropic ADI (IADI) method is proposed based on the (2, 4) stencil, which is more efficient than the explicit IOM method.

C. Difference operators and efficient splitting method

While analyzing ADI-FDTD and CN-based methods, the discrete, node-based field components were used to manipulate the actual update equations for programming, which is time consuming, and easily leads to wrong results. With trial-and-error, it has been found that if the abstract difference operator is used, the manipulation is much easier and the result is usually correct. There is no need to distinguish the difference operators for the electric field components and the magnetic field components if the operators use the difference formula [156]. In addition, since the difference operators can be regarded as common variables, they can be operated with addition, subtraction, multiplication and division; they can also be incorporated into a matrix, and thus can be split differently. Without these simple but powerful "tools", it would have been very difficult to develop the 3D CN-based methods in Chapter 6.

The "efficient-splitting method" [156] presented in Section 5.1 and Section 6.5 is used to avoid the large computational stencils caused by the cross terms in CNDG, CNCS and CNAFS. By the avoidance of large computational stencils, proper material interface treatment is simpler and easier; and the better numerical accuracy is preserved.

D. Efficient implementation of CN scheme

Chapter 4 proposes several methods for efficiently implementing the CN scheme in the 2D case. Originally they were developed intuitively by adding or dropping some terms using the node-based notations [151] [157]. With the difference operators and the efficient-splitting method, they are verified rigorously in Chapter 4. The CNDS and CNAD methods are more efficient than ADI, as indicated by Table 5-1, though they have the same numerical dispersion. For large Courant numbers, they have less RMS discretization error than ADI as shown in Fig. 9-7. CNCS and CNDG can have zero anisotropy for some combinations of mesh density and Courant number. They have higher numerical accuracy than the ADI method in terms of numerical anisotropy. Because there is no associated specific time step, these CN-based methods consume CPU time about the same as or less than ADI. In addition, the coefficient-modification technique is more efficient for CNCS and CNDG than ADI, because they are inherently more isotropic. Numerical experiments have been performed which agree well with the analysis.

Chapter 6 presents several CN-based methods in the 3D case. The CNDS method is more efficient than ADI, as shown in Table 6-1, though CNDS and ADI have the numerical dispersion relation. The CNCSU [163] and CNAFS [156] have much less anisotropy than ADI and CNDS. The CNAFS method is the most suitable for parallel computation and CNCS is suitable for series computation.

The strategy in developing CN-based methods for the first-degree Maxwell's Equations can also be applied in solving for the second-degree wave equations [162], as illustrated in Section 5.7.

For all the unconditionally-stable methods proposed in this thesis, the amplification factor polynomials have been given and the amplification factors have been obtained. The unconditional stability is verified theoretically and also numerically by coding them and running them for various time steps.

E. Perturbation errors

The ADI and CN-based methods are all perturbed versions of the CN scheme. However the perturbation errors are different, and have been derived in this thesis. The lowest order of perturbation error in time for ADI is second order. For CNCS, CNDG and CNAFS is 4^{th} order. Since the CNCS, CNDG and CNAFS better preserve the property of the CN scheme, they are inherently more isotropic. The analysis of perturbation error gives us a hint that 2^{nd} order or higher-order terms can be added to the formulation in order to improve the numerical accuracy.

F. Quantification of discretization errors

In the 2D lossless case, a method to quantify the discretization errors has been proposed in Chapter 8. With the transfer functions for the temporal difference and spatial difference, the error matrices for the explicit methods and unconditionally-stable methods [161] in this thesis are derived by decomposing the plane wave. A general error matrix for explicit FDTD methods has been obtained. The discretization errors for the Yee, O24, NA, and NA24 methods are compared, as shown in Fig. 9-6, and the results are similar to the analysis of numerical dispersion. However, the discretization errors shown in Fig. 9-7

for the unconditionally-stable methods are different from the analysis of numerical dispersion. This is one reason why we develop such a quantifying method. Another reason is to quantitatively compare some schemes for which the numerical dispersion relation is quite complicated, such as 3D CNCS. The discretization errors also show which field component is more isotropic and along which direction a field component is the most accurate. Note that numerical dispersion quantifies phase error, whereas the discretization errors quantify the relative amplitude error.

G. Numerical loss in lossy materials

Chapter 9 points out that in lossy media the loss constant in FDTD is different from its physical value. The relations between numerical phase constant and numerical loss constant have be obtained for Yee's FDTD [164], CN-FETID, CNCS-FDTD and ADI-FDTD, and with good agreement with numerical experiments in the 1D case. With fine mesh densities, the errors of the numerical dispersion and loss tend to be small. For Yee's FDTD, at the Courant limit, both numerical dispersion and loss have higher accuracy which is similar to lossless case, but the errors never vanish; thus no "magic" time step size exists in lossy case. The intuitively-formulated ADI FDTD is first-order accurate in time, and the larger the time step size, the larger the error of the numerical loss. CN-FDTD is second-order accurate in time; its error is much smaller than ADI-FDTD. In particular, the numerical loss error is much smaller than ADI-FDTD. The CN and CN-based methods have higher numerical accuracy than ADI in lossy media.

However, if ADI is written as a perturbation of the original CN scheme instead of a direct discretization of Maxwell's Equations intuitively at each sub-step with unbalanced

time splitting, ADI can have the same accuracy as CN in the 1D case. This is one example of improving FDTD's accuracy after having understood the properties of FDTD.

The relations for the numerical phase and loss constants in higher dimensions have also been derived for Yee [164], CN and CNCS in Section 9.4. Both the numerical constants α and β have anisotropy. The numerical dispersion and loss terms are separated along the diagonals of a square mesh and a cubic mesh and along axes, allowing easy computation of the anisotropies. It has been shown that the numerical accuracy is usually higher along the diagonals than that along the axes, which is similar to the lossless case. But the numerical velocity along the diagonals can be smaller or larger than that along the axes, and the anisotropy of the numerical velocity can be positive or negative. There is a zero-anisotropy of numerical dispersion for certain conductivities and frequencies. However, the numerical velocities along other directions are different from the axes and the diagonal even there is no anisotropy according to the conventional definition Eqn. (3-1). In this special case, the diagonals have a little bit less accurate compared to other direction of travel, though the non-uniformity of the velocity is 10^{-4} which is highly isotropic.

The fact that numerical loss is always larger than its physical value indicates that the FDTD methods *over-estimate* the absorption of electromagnetic energy in lossy media, leading to smaller field strengths than in the physics.

H. Other works

In practice, only some problems with very fine geometric features can be efficiently solved with an unconditionally-stable method. As a guide, Section 7.8 classifies the

problems suitable for application of an USM. A general method to choose a proper time step size for a desired accuracy [155] is given in Section 7.6.

A simple method to solve problems with periodic boundaries is discussed and tested successfully in 2D CNDG and 3D CNAFS. A generalized numerical dispersion relation in the lossless case for the unconditionally-stable methods is given in Eqn. (6-45).

10.3 Future Work

The development of FDTD relies on progress of applied mathematics, physics and electrical engineering, as well as the understanding of FDTD. This thesis has done some preliminary work on developing new methods. However, those methods are by no means unique and final, and leave a lot of room for further improvement.

For the methods proposed in this thesis, future work will first focus on extending the new methods to more general cases, such as complex media, and on developing a high-accuracy PML which is compatible to the methods. Another work is to use these new methods for problems of interest and explore their advantages and weaknesses.

All the methods proposed in this thesis can be extended to the lossy case, and can be modified to overcome the staircase error, and can be incorporated into memory-saving algorithms.

The numerical dispersion relations for 3D CN-based methods in the lossless case are derived in this thesis. Though at present they are theoretically correct, they need further validation from numerical experiments. The discretization error analysis needs to be extended to the 3D case, in which such error quantification is more important because 3D numerical dispersion relation is sometimes difficult to derive; and to lossy materials, in which complex variables will be encountered.

The amplification factors, the numerical dispersion and loss relations for the high-accuracy 3D CNCS, 2D and 3D ADI methods in lossy media, as well as the PML for USM needs further investigation.

Besides, our initial analysis for the 2D equal-lateral triangle mesh [166] shows that it has numerical dissipation though it has very small anisotropy of the numerical dispersion. The investigation needs to be verified with numerical experiments, and to extend it to other irregular meshes commonly used in FDTD methods.

All the new methods proposed in this thesis are based on the Yee's discretization and the step-by-step time marching strategy. It is also possible to devise some new methods with a different strategy, such as the no-time-stepping methods, the neural-network method and a combination of FDTD with a genetic algorithm. In addition, the feed-back control mechanism from the control theory may be borrowed for explicit FDTD to use a time step size larger than the Courant limit where it is used to stabilize some originally-not- stable systems. Because of increasing vitality and popularity of FDTD, new methods with better performance is always desirable. Working on developing such methods is challenging and exciting!

Bibliography

[1] J.M. Jin, "The finite element method in electromagnetics," John Wiley & Sons, Inc., New York, 1993.

[2] W.C. Chew, "Fast and efficient algorithms in computational electromagnetics," Ed.,W.C. Chew, J. Jia, E. Michielssen, J. Song, Artech House, Boston, 2001.

[3] S.M. Rao , "Time domain electromagnetics," Academic Press, San Diego, 1999.

[4] A. Taflove and S.C. Hagness, "Computational electrodynamics—the finite-difference time-domain method," 2nd ed., Artech House, Boston, 2000.

[5] D. Sullivan, "Electromagnetic simulation using the FDTD method," IEEE Press, New York, 2000.

[6] K. S. Kunz and R. J. Luebbers, "The finite difference time domain method for electromagnetics," CRC Press, Boca Raton, FL., 1993.

[7] D.H. Werner and R. Mittra, "Frontiers in electromagnetics," IEEE Press, New York, 2000.

[8] M.V.K. Chari and S.J. Salon, "Numerical methods in electromagnetism," Academic Press, San Diego, 2000.

[9] A. Bossavit, "Computational electromagnetics —variational formulations, complementarty, edge elements," Academic Press, San Diego, 1998.

[10] A. Taflove, "Advances in computational electrodynamics—the finite-difference time-domain method," Artech House, Boston, 1998

[11] M.N.O. Sadiku, "Numerical techniques in electromagnetics," 2nd ed., CRC Press, New York, 2001

[12] M. Levy, "Parabolic equation methods for electromagnetic wave propagation," The Institution of Electrical Engineers, London, 2000.

[13] R.W. Ziokowski and A.D. Kipple, "Causulty and double-negative metamaterials," Phys. Rev. E, Vol. 68, E026615, 2003.

[14] K.S. Yee, " Numerical solution of initial boundary value problems involving Maxwell's Equations in isotropic media," IEEE Trans. Antennas Propagat., Vol. 14, pp. 302-307, May 1966.

[15] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic field problems", IEEE Trans. Electromagn. Compat., Vol. EC-23, No. 4, pp. 377-382, Nov. 1981.

[16] K.J. Bowers, "Implicit methods of solving the Maxwell's Equations suitable for particle-in-cell simulation of low temperature plasmas," Available at http://ptsg.eecs.berkeley.edu/~kbowers/articles.html

[17] A. Ahland, D. Schuulz, and E. Voges, "Accurate mesh truncation for Schrönding equations by a perfectly matched layer absorber: Applications to the calculation of optical spectra," Phys. Rev. B, Vol. 60, No. 8, pp. 5109-5112, Aug. 1999.

[18] G.C. Cohen, "Higher-order numerical methods for transient wave equations," Springer, New York, 2002.

[19] D. Moreloose, J.T. Dawson, and M.A. Stuchly, "Applications of the finite difference time domain algorithm to quasi-static field analysis," Radio Science, Vol. 2, No. 2, pp. 329-341, Mar.-Apr. 1997.

[20] I.J. Craddock and C.J. Railton, "A new technique for the stable incorporation of static field solutions in the FDTD method for the analysis of thin wires and narrow strips", IEEE Trans. Microwave Theory Tech., Vol. 46, pp. 1091-1096, Aug. 1998.

[21] M.S. Sarto, "Innovative absorbing-boundary conditions for the efficient FDTD analysis of lighting-interaction problems," IEEE Trans. Electromagn. Compat., Vol. 43, pp368-381, Aug. 2001.

[22] C.R. Paul, "Introduction to electromagnetic compatibility," Wily, New York, 1992.

[23] N. Chavannes, R. Tay, N. Nikoloski, and N. Kuster, "Suitability of FDTD-based TCAD tools for RF design of mobile phones," IEEE Antennas Propagt. Mag., Vol. 45, pp. 52-66, Dec. 2003.

[24] M. Krumpholz and L.P.B. Katehi, "MRTD: New time-domain schemes based on multi-resolution analysis," IEEE Trans. Microwave Theory Tech., Vol. 44, pp. 555-571, Apr. 1996.

[25] Q.H. Liu, "The PSTD method: A time-domain method requiring only two cells per wavelength," Microwave Opt. Tech. Lett., Vol. 15, No. 3, pp.158-165, Jun. 1997.

[26] J. Fang, "Time domain finite difference computation for Maxwell's Equations," Ph.D dissertation, Univ. of California, Berkeley, CA, 1989

[27] M.F. Hadi and M. Piket-May, "A modified FDTD (2, 4) scheme for modeling large structures with high-phase accuracy," IEEE Trans. Antennas Propagat., Vol. 45, pp. 254-264, Feb. 1997.

[28] K. Lan, Y. Liu, and W. Lin, "A higher order (2,4) scheme for reducing dispersion in FDTD algorithm," IEEE Trans. Electromagn. Compat., Vol. 41, pp. 160- 165, May 1999.

[29] A. Yefet and P.G. Petropoulos, "A staggered fourth-order accurate explicit finite difference scheme for the time-domain Maxwell's Equations," J. Comput. Phys., Vol. 168, No. 2, pp. 286-315, Apr. 2001.

[30] H.E. Abd El-Raouf, E.A. El-Diwani, A. El-Hadi, and F.M. El-Hefnawi, "A low-dispersion 3D second-order in time forth-order in space FDTD scheme (M3d24)," IEEE Trans. Antennas Propagat., Vol. 52, pp. 1638-1646, Jul. 2004.

[31] Z. Xie, C.-H. Chan, and B. Zhang, "An explicit fourth-order staged finite-difference time-domain method for Maxwell's Equations," J. Compt. Appl. Math., Vol. 147, No. 1, pp. 75-98, Oct. , 2002.

[32] Z. Shao, Z. Shen, Q. He, and G. Wei, "A generalized higher order finite-difference time-domain method and its application in guided-wave problems," IEEE Trans. Microwave Theory Tech., Vol. 51, pp. 856-861, Mar. 2003.

[33] D.W. Zingg, "Comparison of higher-accuracy finite difference methods for linear wave propagation," SIAM J. Sci. Comput., Vol. 22, No. 2, pp. 476-502, Jul. 2000.

[34] K.L. Shlager and J.B. Schneider, "Comparison of the dispersion properties of several low-dispersion finite-difference time-domain algorithms," IEEE Trans. Antennas Propagat., Vol. 51, pp. 642-653, Mar. 2003.

[35] K.L. Shlager and J.B. Schneider, "Comparison of the dispersion properties of higher order FDTD schemes and equivalent-sized MRTD schemes," IEEE Trans. Antennas Propagt., Vol. 54, pp. 1095-1104, Apr. 2004.

[36] J.S. Juntunen and T.D. Tsiboukis, "Reduction of numerical dispersion in FDTD method through artificial anisotropy", IEEE Trans. Microwave Theory Tech., Vol. 48, pp. 582-588, Apr. 2000.

[37] M. Rewienski and M. Mrozowski, "An iterative algorithm for reducing dispersion error on Yee's mesh in cylindrical coordinates", IEEE Microwave Guided Wave Lett., Vol. 10, pp. 353-355, Sept. 2000.

[38] M. Wang, Z. Wang, and J. Chen, "A parameter optimized ADI-FDTD method," IEEE Antennas Wireless Propagt. Lett., Vol. 2, No. 8, pp. 118-121, 2003.

[39] J.W. Nehrbass, J.O. Jevtic, and R. Lee, "Reducing the phase error for finite-difference methods without increasing the order," IEEE Trans. Antennas Propagat., Vol. 46, pp. 1194-1201, Aug. 1998.

[40] J.B. Cole, "high accuracy solution of Maxwell's Equations using nonstandard finite differences," J. Comput. Phys., Vol. 11, pp. 287-292, May 1997.

[41] J.B. Cole, "High-accuracy Yee algorithm based on nonstandard finite differences: New development and validations," IEEE Trans. Antennas Propagat., Vol. 50, pp. 1185-1191, Sept. 2002.

[42] T.T. Zygiridis and T.D. Tsiboukis, "A dispersion-reduction scheme for the higher order (2,4) FDTD method," IEEE Trans. Magnetics, Vol. 40, pp. 1464-1467, Mar. 2004.

[43] T.T. Zygiridis and T.D. Tsiboukis, "Low dispersion algorithms based on the higher order (2, 4) FDTD method," IEEE Trans. Microwave Theory Tech., Vol. 52, pp.1321-1327, Apr. 2004.

[44] E.A. Forgy and W.C. Chew, "A time-domain method with isotropic dispersion and increased stability on an overlapped lattice," IEEE Trans. Antennas Propagt., Vol. 50, pp. 983-996, Jul. 2002.

[45] S. Wang and F.L. Teixeira, "A three-dimensional angle-optimized finite-difference time-domain algorithm," IEEE Trans. Microwave Theory Tech., Vol. 51, pp. 811-817, Mar. 2003.

[46] S. Wang and F.L. Teixeira, "A finite-difference time-domain algorithm optimized for arbitrary propagation angles," IEEE Trans. Antennas Propagat., Vol. 51, pp. 2456-2463, Sept. 2003.

[47] S. Wang and F.L. Teixeira, "Dispersion-relation-preserving FDTD algorithms for large-scale three-dimensional problems," IEEE Trans. Microwave Theory Tech., Vol. 51, pp.1818-1828, Aug. 2003.

[48] S. Wang and F.L. Teixeira, "Grid-Dispersion Error Reduction for Broadband FDTD Electromagnetic Simulations," IEEE Trans. Magnetics, Vol. 40 , No. 2 , pp. 1440-1443, Mar. 2004.

[49] G.D. Smith, "Numerical solution of partial differential equations: Finite difference methods," 3rd ed., Clarendon Press, Oxford University Express, Oxford, 1985.

[50] M. Mrozowski, "Stability condition for the explicit algorithms of the time domain analysis of Maxwell's Equations," IEEE Microwave Guided Wave Lett., Vol. 4, pp. 279-281, Aug. 1994.

[51] R.F. Remis, "On the stability of the finite-difference time-domain method," J. Comput. Phys., Vol. 163, No. 1, pp. 249-261, Sept. 2000.

[52] J.A. Pereda, L.A. Vielva, A. Vegas, and A. Prieto, "Analyzing the stability of the FDTD technique by combining the Von Neumann method with Routh-Hurwitz Criterion" IEEE Trans. Microwave Theory Tech., Vol. 49, pp.377-381, Feb. 2001.

[53] B.Y. Scobelev and E.V. Vorozhtsov, "Sufficient stability criteria and uniform stability of difference schemes," J. Compt.Phys., Vol. 165, No. 2, pp. 717-751, Dec. 2000.

[54] S.S. Abarbanel and A.E. Chertock, "Strict stability of high-order compact implicit finite-difference schemes: The role of boundary conditions for hyperbolic PDEs," Part I, J. Comput. Phys., Vol. 160, No. 1, pp. 42-66, May 2000.

[55] C.T. Schroder and W.R. Scott, Jr., "On the stability of the FDTD algorithm for elastic media at a material interface," IEEE Tran. Geosci. Remote Sensing, Vol. 40, pp. 474-481, Feb. 2002.

[56] G. Marrocco and F. Bardati, " Time-domain macromodel of planar microwave devices by FDTD and moment expansion," IEEE Trans. Microwave Theory Tech., Vol. 49, pp. 1321-1327, Jul. 2001.

[57] M. Okoniewski, E. Okoniewski, and M.A. Stuchly, "Three-dimensional subgridding algorithm for FDTD," IEEE Trans. Antennas Propagat., Vol. 45, pp. 422-428, Mar. 1997.

[58] A. Monorchio and R. Mittra, "A novel subgridding scheme based on a combination of the finite-element and finite-difference time-domain methods," IEEE Trans. Antennas Propagat., Vol. 46, No. 9, pp. 1391-1393, Sept. 1998.

[59] V. Nefedov, "Subgridding in FDTD," 15th international Conference on Domain Decomposition Methods, July 21-25, 2003, Berlin, Germany, Technical Report RANA 02-29, TU Eindhoven, Eindhoven, 2002.

[60] B. Denecker, F. Olyslager, L. Knockaert, and D.D. Zutter, " Automatic generation of subdomain models in 2-D FDTD using reduced order modeling," IEEE Microwave Guided Wave Lett., Vol. 10, pp. 301-303, Aug. 2000.

[61] S. Ju, K.-Y. Jung and H. Kim, " Investigation on the characteristics of the envelope FDTD based on the alternating direction implicit scheme," IEEE Microwave Wireless Components Lett., Vol. 13, pp. 414 – 416, Sept. 2003.

[62] H. Rao, R. Scarmozzino, and R.M. Osgood, Jr., "An improved ADI-FDTD method and its application to photonic simulations," IEEE Photon. Technol. Lett., Vol. 14, pp. 477-479 , Apr. 2002.

[63] C. Ma and Z. Chen, "Stability analysis of the CE-FDTD method," IEEE Microwave Wireless Components Lett., Vol. 14 , pp. 243- 245, Mar. 2004

[64] C. Wu, E.A. Navarro, J. Navasquillo, and J. Litva, "FDTD signal extropolation using a finite impulse response neural network model," Microwave & Opt. Tech. Letts., Vol. 21, No. 5, pp. 325-330, Jun. 1999

[65] A. Taflove, "Numerical issues regarding finite-difference time-domain modeling of microwave structures," in "Time-domain methods for microwave structures: Analysis and Design," T. Itoh, B. Houshmand, ed., IEEE Press, New York, 1998.

[66] H. Kawaguchi, K. Takahara, and D. Yamauchi, "Design study of ultra-speed mivrowave simulator engine," IEEE Trans. Magnetics, Vol. 38, No. 2, pp. 689-692, Mar. 2002.

[67] J.P. Durbano, F.E. Ortiz, J.P. Humphrey, M.S. Mirotznik and D.W. Prather, "Hardware implementation of a three-dimensional fininte-difference time-domain algorithm," IEEE Antennas Wireless Propagat. Lett., Vol. 2, pp. 54-57, 2003.

[68] Y. Lu and C.Y. Shen, "A domain decomposition finite-difference method for parallel numerical implementation of time-dependent Maxwell's Equations," IEEE Trans. Antennas Propagat., Vol. 45, pp. 556-562, Mar. 1997.

[69] J.C. Strikwerda, " Finite difference schemes and partial differential equations," Wadawoth & Brooks/Cole Advanced Books &Software, Pacific Grove, California, 1989.

[70] A.C. Cangellaris and L. Zhao, "Rapid FDTD simulation without time stepping," IEEE Microwave Guided Wave Lett., Vol. 9, pp. 4-6, Jan. 1999.

[71] Y.-S. Chung, T.K. Sarkar, B.H. Jung, and M. Salazar–Palma, "A unconditionally stable scheme for the finite-difference time-domain method," IEEE Trans. Microwave Theory Tech., Vol. 51, pp. 697-704,, Mar. 2003.

[72] H.D. Raedt, K. Michielsen, J.S. Kole, and M.T. Figge, " Solving the Maxwell's Equations by the Chebyshev method: A one-step finite-difference time-domain algorithm" IEEE Trans. Antennas Propagat., Vol. 51pp. 3155 - 3160, Nov. 2003.

[73] J.S. Kole, M.T. Figge, and H.D. Raedt, "Unconditionally stable algorithms to solve the time-dependent Maxwell's Equations," Phys. Rev. E, Vol. 64, E066705, Nov. 2001.

[74] J.S. Kole, M.T. Figge, and H.D. Raedt, "Higher-order unconditionally stable algorithms to solve the time-dependent Maxwell's Equations," Phys. Rev. E, Vol. 65, E 066705, Jun. 2002.

[75] B. Gustafsson, H.-O. Kreiss, and J. Oliger, "Time dependent problems and difference methods," New York, John Wiley & Sons, Inc., 1995.

[76] K.W. Morton and D.F. Mayers, "Numerical solution of partial differential equations," Cambridge University Press, Cambridge, England, 1994.

[77] J.W. Thomas, "Numerical partial differential equations," Springer-Verlag, Inc. , New York, 1998.

[78] A.R. Mitchell, "Computational methods in partial differential equations, " John Wiley & Sons, Inc., New York, 1969.

[79] S. Doss and K. Miller, "Dynamic ADI methods for elliptic equations, " SIAM J. Numer. Anal., Vol.16, No. 5, pp. 837-847 , Oct. 1979.

[80] R. Holland, "Implicit three-dimensional finite differencing of Maxwell's Equations for computational electromagnetism," IEEE Trans. Nucl. Sci., Vol. 31, pp. 1322-1326, Dec. 1984.

[81] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," IEEE Trans. Microwave Theory Tech., Vol. 47, pp. 2003 -2007, Oct. 1999.

[82] T. Namiki, "3-D ADI-FDTD method-unconditionally stable time-domain algorithm for solving full vector Maxwell's Equations," IEEE Trans. Microwave Theory Tech., Vol. 48, pp. 1743 –1748, Oct. 2000.

[83] 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 , pp. 1550 –1558, Sept. 2000.

[84] S.W. Staker, C.L. Holloway, A.U. Bhobe, and M. Piket-May, "Alternating-direction implicit (ADI) formulation of the finite-difference time-domain (FDTD) method:

algorithm and material dispersion implementation," IEEE Trans. Electromagn. Compat., Vol. 45, pp. 156-165, May 2003.

[85] C.C.-P. Chen, T.-W. Lee, N. Murugesan, and S.C. Hagness, "Generalized FDTD-ADI: an unconditionally stable full-wave Maxwell's Equations solver for VLSI interconnect modeling ," IEEE/ACM International Conference on Computer Aided Design, ICCAD-2000, pp. 156-163, San Jose, 5-9, Nov. 2000.

[86] C. Yuan and Z. Chen, "On the modeling of conducting media with the unconditionally stable ADI-FDTD method," IEEE Trans. Microwave Theory Tech., Vol. 51, pp. 1929-1938, Aug. 2003.

[87] S.G. Garcia, R.G. Rubio, A.R. Bretones, and R.G. Martin, "Extension of the ADI-FDTD method to Debye media," IEEE Trans. Antennas Propagat., Vol. 51, pp. 3183-3186, Nov. 2003.

[88] B. Wang, Y. Wang, W. Yu, and R. Mittra, "A hybrid 2-D ADI-FDTD subgriding scheme for modeling on-chip interconnects," IEEE Trans. Adv. Packag., Vol. 24, pp. 528-533, Nov. 2001.

[89] I.A. Hmed and Z. Chen, "A hybrid FDTD and ADI-FDTD scheme for RF/Microwave structure simulation," the 19th Annual Review of Progress in Applied Computational Electromagnetics, pp. 408-412, Monterey, CA, 24-28, Mar. 2003.

[90] Z. Wang, F. Liu and J. Chen, "Analysis and application of compact form ADI-FDTD method to substrate and interconnect simulation," 2003 IEEE International Symposium on Electromagnetic Compatibility, Vol. 2, pp. 585-590, Boston, MA, 18-22, Aug. 2003.

[91] S. Ju and H. Kim, "Investigation of an unconditionally stable compact 2D ADI-FDTD algorithm: formulations, numerical stability, and numerical dispersion," Digest 2002 IEEE Antennas and Propagation Society International Symposium, Vol. 3, pp. 639-642, San Antonio, TX, 16-21, Jun. 2002

[92] C. Yuan and Z. Chen, "A three-dimensional unconditionally stable ADI-FDTD method in the cylinderical coordinate system," IEEE Trans. Microwave Theory Tech., Vol. 50, pp. 2401-2405, Oct. 2002.

[93] P. Pongpaibool, A. Kamo, T. Watanabe, and H. Asai, "An alternating implicit block overlapped FDTD (AIBO-FDTD) method and its estimation with parallel computation, " IEEE Electrical Performance of Electronic Packaging, pp. 185 – 188, Cambridge, MA, 29-31, Oct. 2001.

[94] F. Zheng and Z. Chen, " A low-dispersive high-order unconditionally stable FDTD method ," IEEE Antennas and Propagation Society International Symposium, Vol. 3, pp. 1514 -1517, Salt Lake City, Utah, 16-21, Jul. 2000.

[95] M.K. Sun and W.Y. Tam, "An unconditionally stable high-order 2-D ADI FDTD method," IEEE Antennas and Propagation Society International Symposium, Vol. 4, pp. 352 – 355, Hong Gong, China, 22-27, Jun. 2003.

[96] J. Chen, Z. Wang and Y. Chen, "Higher-order alternative direction implicit FDTD method," IEE Electron. Lett., Vol. 38 , No. 22 , pp. 1321 – 1322, Oct. 2002.

[97] N.V. Kantartzis, T.T. Zygiridis, and T.D. Tsiboukis, "An unconditionally stable higher order ADI-FDTD technique for the dispersionless analysis of generalized 3-D EMC structures," IEEE Trans. Magnetics, Vol. 40, pp. 1436-1439, Mar. 2004.

[98] A. Zhao, "Improvement on the numerical dispersion of 2D ADI-FDTD with artificail anisotorpy," IEEE Microwave Wireless Components Lett., Vol. 14, pp. 292-294, 2004.

[99] F. Zheng and Z. Chen , "Numerical dispersion analysis of the unconditionally stable 3-D ADI-FDTD method," IEEE Trans. Microwave Theory Tech.,Vol. 49, pp. 1006-1009, May 2001.

[100] A. Zhao, "Analysis of the numerical dispersion of the 2D alternating-direction implicit FDTD method," IEEE Trans. Microwave Theory Tech., Vol. 49, pp. 1006-1009, May 2001.

[101] M. Darms, R. Schuhmann, H. Spachmann, and T. Weiland, "Dispersion and asymmetry effects of ADI-FDTD," IEEE Microwave Wireless Components Lett., Vol. 12, pp. 491-493, Dec. 2002.

[102] Z. Wang and J. Chen, "Analytical and numerical accuracy analysis of ADI-FDTD methods" IEEE Antennas and Propagation Society International Symposium, Vol. 4 , 22-27, Jun. 2003.

[103] S. Ju, H. Kim, and H. -H., Kim, "A study of the numerical dispersion relation for the 2D ADI-FDTD method," IEEE Microwave Wireless Components Lett., Vol. 13, pp. 405-407, Sept. 2003.

[104] S.G. Garcia, T.-W. Lee, and S.C. Hagness, "On the accuracy of the ADI-FDTD method," IEEE Antennas Wireless Propagat. Lett. , Vol. 1 , pp. 31 –34, 2002.

[105] A. Fijany, M.A. Jensen, Y. Rahmat-Samii and J. Barhen, "A massively parallel computation strategy for FDTD: Time and space parallelism applied to electromagnetics problems," IEEE Trans. Antennas Propagat., Vol. 43, pp. 1441-1449, Dec. 1995.

[106] B. Fornberg, "Some numerical techniques for Maxwell's Equations in different types of geometries," Topics in Computational Wave Propagation 2002, Springer, 2003.

[107] J. Lee and B. Fornberg, "Some unconditionally stable time stepping methods for the 3D Maxwell's Equations," J. Computat. Appl. Math., Vol. 166, No. 2, pp. 497-523, Apr. 2004.

[108] J.S. Shang, "A fractional-step method for solving 3D, time-domain Maxwell's Equations," J. Comput. Phys., Vol. 118, No. 1, pp. 109-119, Apr. 1995.

[109] J.S. Shang and M.F. Robet, "A comparative study of characteristic-based algorithms for Maxwell's Equations," J. Compt. Phys., Vol. 125, No. 2, pp. 378-394, Apr. 1996.

[110] J.H. Beggs and W.R. Briley, "An implicit characteristic based method for electromagnetics," NASA/TM-2001-210862, May 2001

[111] W. Yu, R. Mittra, and S. Dey, "Application of the nonuniform FDTD technique to analysis of coaxial discontinuity structures," IEEE Trans. Microwave Theory Tech., Vol. 49, pp. 207-209, Jan. 2001.

[112] T. Xiao and Q.H. Liu, "A staggered upwind embedded boundary (SUEB) method to eliminate the FDTD staircasing error," IEEE Trans. Antennas Propagat., Vol. 52, pp. 730-741, Mar. 2004.

[113] P.H. Aoyagi, J. Lee, and R. Mittra, " A hybrid Yee algorithm/scalar-wave equation approach," IEEE Trans. Microwave Theory Tech., Vol. 41, pp. 1593-1600, Sept. 1993.

[114] F.D. Faviis, M. Noro, R. E. Diaz, G. Franceschetti, and N.G. Alexopoulos, " A vector potential formulation for the solution of electromagnetic problems," IEEE Microwave Guided Wave Lett., Vol. 8, pp. 310-312, Sept. 1998.

[115] G.D. Kondylis, F.D. Faviis, G.J. Pottie and T. Itoh, "A memory-efficient formulation of the finite-difference time-domain method for the solution of Maxwell's Equations," IEEE Trans. Microwave Theory Tech., Vol. 49, pp. 1310-1320, Jul. 2001.

[116] J. Vazquez and C.G. Parini, "Antenna modeling using discrete Green's function formulation of FDTD method," IEE Electron. Lett., Vol. 35, No. 13, pp. 1033-1034, Jun. 1999.

[117] R. Holtzman and R. Kastner, "The time-domain discrete Green's function method (GFM) characterizing the FDTD grid boundary," IEEE Trans. Antennas Propagat., Vol. 49, pp. 1079-1093, Jul. 2001.

[118] Z.P. Liao, H.L. Wong, B.P. Yang, and Y. F. Yuan, "A transmitting boundary for transient wave analysis"" Sci. Sinica, Vol. XXVII, No. 10, pp. 1063-1076, Oct. 1984.

[119] 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.

[120] J.A. Pereda, O. Garcia, A. Vegas and A. Prieto, " Numerical dispersion and stability analysis of the FDTD technique in lossy dielectrics," IEEE Microwave Guided Wave Lett., Vol. 8, pp. 245-247, Jul.1998

[121] W.A. Beck and M.S. Mirotznik, " Generalized analysis of the stability and numerical dispersion in the discrete-convolution FDTD method," IEEE Trans. Antennas Propagat., Vol. 48, pp. 887-894, Jun. 2000.

[122] J.B. Schneider and R.J. Kruhalk, "Dispersion of homogeneous and inhomogeneous waves in the Yee finite-difference time-domain grid", IEEE Trans. Microwave Theory Tech., Vol. 49, pp. 280-287, Feb. 2001.

[123] J.A. Pereda, A. Vegas, and A. Prieto, "FDTD modeling of wave propagation in dispersive media by using the Mobius transformation technique," IEEE Trans. Microwave Theory Tech., Vo.50, pp. 1689-1695, Jul. 2002.

[124] K.P. Prokopidis and T.D. Tsiboukis, "Higher-order FDTD(2,4) scheme for accurate simulation in lossy dielectrics," IEE Electron. Lett., Vol. 39, pp. 835-836, May 2003.

[125] R. Vichnevetsky, "Elliptic equations and the finite-element method," in "Computational Methods for Partial Differential Equations," Vol. 1, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[126] P.G. Ciarlet and J.L. Lions, "Finite difference methods (part 1)," in "Handbook of Numerical analysis," Amsterdam, North-Holland, 1990.

[127] W.F. Ames, "Numerical methods for partial differential equations," 3rd Ed., Academic Press, Boston, 1992.

[128] Y. Liu, "Fourier analysis of numerical algorithms for the Maxwell's Equations," J. Comput. Phys., Vol. 124, No. 2, pp. 396-416, Mar. 1996.

[129] V.G. Ganzha and E.V. Vorozhtsov, "Computer-aided analysis of difference schemes for partial differential equations," John Willy & Sons, Inc., New York, 1996.

[130] G.E. Forsythe and W.R. Wason, "Finite-difference methods for partial differential equations," John Wiley & Sons, Inc., New York, 1960 .

[131] R. N. Bracewell, "The Fourier transform and its applications," 3rd ed., McGraw Hill, Boston, 2000.

[132] M.S. Min and C.H. Teng, "The instability of the Yee scheme for the 'magic time step', " J. Comput. Phys., Vol. 166, No. 2, pp. 418-424, Jan. 2001.

[133] M.H. Carpenter, D. Gottlieb, and S. Abarbanel, "Time-stable boundary conditions for finite difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes," J. Comput. Phys., Vol. 111, No. 2, pp. 220-236, Apr. 1994.

[134] R.M. Beam, R.F. Warming, and H.C. Yee, "Stability analysis of numerical boundary conditions and implicit difference approximation for hyperbolic equations," J. Comput. Phys., Vol. 48, No. 2, pp. 200-222, Nov. 1982.

[135] C.A. Balanis, "Advanced engineering electromagnetics", New York: John Wiley & Sons, Inc., 1989.

[136] J.B. Schneider, C.L. Wagner, and O.M. Ramahi, "Implementation of transparent sources in FDTD simulations," IEEE Trans. Antennas Propagat., Vol. 46, pp. 1159-1168, Aug. 1998.

[137] R. Pontalti, J. Nadobny, P. Wust, A. Vaccari and D. Sullivan, "Investigation of static and quasi-static fields inherent to the pulsed FDTD method," IEEE Trans. Microwave Theory Tech., Vol. 50, pp. 2022-2025, Aug. 2002.

[138] C.M. Furse, D.H. Roper, D.N. Buechler, D.A. Christensen and C.H. Durney, "The problem and treatment of DC offsets in FDTD simulations," IEEE Trans. Antennas Propagat., Vol. 48, pp. 1198-1201, Aug. 2000.

[139] B.C. Kuo and F. Golnaraghi, "Automatic control systems," 8th Ed., John Wiley & Sons, Inc., New York, 2003.

[140] T. Martin, "An improved near-to-far-zone transformation for the finite-difference time-domain method," IEEE Trans. Microwave Theory Tech., Vol. 46, pp. 1263-1271, Sept. 1998.

[141] J. Visher, S. Wandzura, and A. White, "Stable, higher order discretization for evolution of the wave equations in 1+1 dimensions," J. Compt. Phys., Vol. 194, No. 2, pp. 395-408, Feb. 2004.

[142] H.-O Kreiss, N.A. Petersson and J. Ystrom, "Difference approximations for the second order wave equations," SIAM J. Numer. Anal., Vol. 40, pp. 1940-1967, No. 5, 2002.

[143] P. Ricci, G. Lapenta and J.U. Brackbill, "A simplified implicit Maxwell solver," J. Comput. Phys., Vol. 183, No. 1, pp. 117-141, Nov. 2000.

[144] J.-L. Vay, "An extended FDTD scheme for the wave equation: application to multiscale electromagnetic simulation," J. Comput. Phys., Vol. 167, No. 1, pp. 72-98, Feb. 2001.

[145] N.J. Higham "Stability of block LDL^T factorization of a symmetric tridiagonal matrix," Linear Algebra and its Applications, Vol. 287, No. 1-3, pp. 181-189, Jan. 1999.

[146] A. Zhao, "The influence of the time step on the numerical dispersion error of an unconditionally stable 3-D ADI-FDTD method: a simple and unified approach to determine the maximum allowable time step required by a desired numerical dispersion accuracy," Microwave & Opt. Tech. Lett., Vol. 35, pp. 60-65, Oct. 2002.

[147] B. Huang, G. Wang, Y. Jiang, and W. Wang, "A hybrid implicit-explicit FDTD scheme with weakly conditional stability," Microwave & Opt. Tech. Lett., Vol. 39, No. 2, pp. 97-101, Oct. 2003.

[148] R. Schuhmann and T. Weiland, "Stability and conservation properties of transient field simulations using FIT," Advances in radio Science, Vol. 1, pp. 93-97, May 2003.

[149] E. Blayo, "Compact finite difference schemes for ocean model," J. Comput., Phys., Vol. 164, No. 1, pp. 241-257, Nov. 2000.

[150] A. Zhao, "A novel implementation for two-dimensional unconditionally stable FDTD method," Microwave & Opt. Tech. Lett., Vol. 38, pp457-462, Sept., 2003.

[151] G. Sun and C.W. Trueman, "Unconditionally stable Crank-Nicolson scheme for solving the two-dimensional Maxwell's Equations," IEE Electron. Lett., Vol. 39, pp. 595-597, Apr. 2003.

[152] G. Sun and C.W. Trueman, "Analysis and Numerical Experiments on the Numerical Dispersion of Two-Dimensional ADI-FDTD," IEEE Antennas Wireless Propagat. Lett., Vol. 2, pp. 78-81, May 2003.

[153] G. Sun and C.W. Trueman, "Accuracy of Three Unconditionally-Stable FDTD Schemes for Solving Maxwell's Equations," Applied Computational Electromagnetics Society Journal, Vol. 18, No. 4, pp. 41-47, Nov. 2003.

[154] G. Sun, C.W. Trueman, "Some fundamental characteristics of the alternate-direction-implicit finite-difference time-domain method," IEEE Trans. Microwave Thoery Tech., Vol. 52, pp. 46-52, Jan. 2004.

[155] G. Sun and C.W. Trueman, "A simple method to determine the time step size to achieve a desired dispersion accuracy in ADI-FETID," Microwave & Opt. Tech. Lett., Vol. 40, No. 6, pp. 487-490, 20, Mar. 2004.

[156] G. Sun and C.W. Trueman, "An unconditionally-stable FDTD method based on the Crank-Nicolson scheme for solving the three-dimensional Maxwell's Equations," IEE Electron. Lett., Vol. 40, No. 10, pp. 589-590, May 2004.

[157] G. Sun and C.W. Trueman, "Approximate Crank-Nicolson schemes for the 2D finite-difference time-domain method in solving electric field components of TE_z wave" , to be published on IEEE Trans. Antennas Propagat., Nov. 2004.

[158] G. Sun and C.W. Trueman, "Numerical validation of dispersion relations using a cylindrical wave for 2D FDTD methods," Microwave & Opt. Tech. Lett , Vol. 43, No. 2, pp. 138-142, 20, Oct. 2004.

[159] G. Sun and C.W. Trueman, "Optimized Finite-Difference Time-Domain Methods", to be published on IEEE Trans. Microwave Theory and Technology.

[160] G. Sun and C.W. Trueman, "Numerical Dispersion in the Alternate-Direction-Implicit Finite-Difference Time-Domain Method," The 19th Annual Review of Progress in Applied Computational Electromagnetics, Monterey, California, 24-28, Mar. 2003.

[161] G. Sun and C.W. Trueman, "Quantification of the truncation errors in finite-difference time-domain methods," CCECE/TEEE 2003-Canadian Conference on Electrical and Computer Engineering and Humane Technology, Montreal, Canada, 4-7, May 2003

[162] G. Sun and C.W. Trueman, "Unconditionally stable solution for wave equations of the second order," ISAPE'03, IEEE/CIE, 6th International Symposium on Antennas, Propagations and EM Theory, Beijing, China, Oct. 28- Nov. 1, 2003.

[163] G. Sun and C.W. Trueman, "The unconditionally-stable cycle-sweep method for 3D FDTD ," ANTEM04, 10th International Symposium on Antenna Technology and Applied Electromagnetics and URSI Conference, Ottawa, Canada, 21-23, Jul. 2004.

[164] G. Sun and C.W. Trueman, "Numerical dispersion and numerical loss in explicit finite-difference time-domain methods in lossy media," submitted for review to the Transactions on IEEE Antennas and Propagation.

[165] G. Sun, "Alternate-direction implicit finite-difference time-domain method," ENCS 8011, ECE Dept., Concordia University, Feb. 2002.

[166] G. Sun, "Analysis and control of numerical dispersion and loss in finite-difference time-domain algorithms," ENCS 851, ECE Dept., Concordia University, Nov. 2002.

[167] G. Sun and C. W. Trueman, "Solutions of Crank-Nicolson scheme of finite-difference time-domain method ," ECE Dept., Concordia University, Aug. 2003.