Numerical Simulation of a Micro-ring Resonator with Adaptive Wavelet Collocation Method

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN

von der Fakultät für Mathematik des

Karlsruher Institut für Technologie

genehmigte

DISSERTATION

von

M. Sc. Haojun Li
aus Daejeon, South Korea

Tag der mündlichen prüfung: 13.07.2011
Referent: Prof. Dr. Andreas Rieder
Korreferent: Prof. Dr. Christian Wieners

Contents

Introduction

0.1 Introduction to the problem to be simulated

Micro-ring resonator is an optical device which consists of a circular ring cavity in the center and coupled by two separated straight waveguides through an air gap of a few hundred nanometers. Optical signals which are imported from one of the straight waveguides can be resonated into the ring cavity and again be switched to another straight waveguide if their frequencies match. People from industry are interested in designing of micro-ring resonators. Such optical devices are useful components for wavelength filtering, switching, routing [16, 35]. Due to huge cost of material based experiments, numerical simulations have become indispensable approaches.

Mathematical problem of micro-ring resonator is nothing but to solve time domain Maxwell's equations. A method called *finite difference time domain* (FDTD) [43] has been used to various types of application problems involving time domain Maxwell's equations, including numerical simulations of micro-ring resonators [16]. To represent the localized fields with high accuracy, FDTD has to sacrifice a large number of numerical grid points even in the region where the requirement of fields resolution is relatively low. Hence, another method called adaptive wavelet collocation method (AWCM) which dynamically adjusts the distribution of numerical grid points is motivated.

0.2 Motivation of AWCM

Assume there is a 1D Gaussian Pulse propagating towards the positive direction of x-axis (Figure 1). To represent the signal numerically, one has to use certain number of points around the peak; however, the amount of points with same density is unnecessary in the region far away from the peak, at least before the peak approaches there. Thus, a more effective way of distributing computational grid points is needed. The distribution should not be uniform but nonuniform and should dynamically change as the peak moves to the right.

When we consider adaptivity of numerical grid points, we have two aspects: first, some parts of the numerical grid points in the current time step may become less important in the next time step, and should be discarded; second, some other parts of the numerical grid points may become significant, thus, more points should be added to that region. In every time step, we perform throwing away and adding some more of grid points. Wavelets which describe detail information of different resolution levels of a function can be a straightforward way of deciding the distribution of numerical grid points effectively. In other words, using wavelet for adaptivity strategy is a natural choice.

Especially for time evolutionary equations, an effective method called *adaptive wavelet collocation method* (AWCM) ([40], [39], [30], [20], [41] and [42], etc.) has been developed and verified. In this thesis, we investigate the applicability of AWCM to solve the time domain Maxwell's equations numerically which is also one system of the evolutionary equations, and compare the results of numerical simulations with other methods, such as FDTD, *interpolating scaling functions method* (ISFM) [14], *Coupled Mode Theory* (CMT) [17], etc .

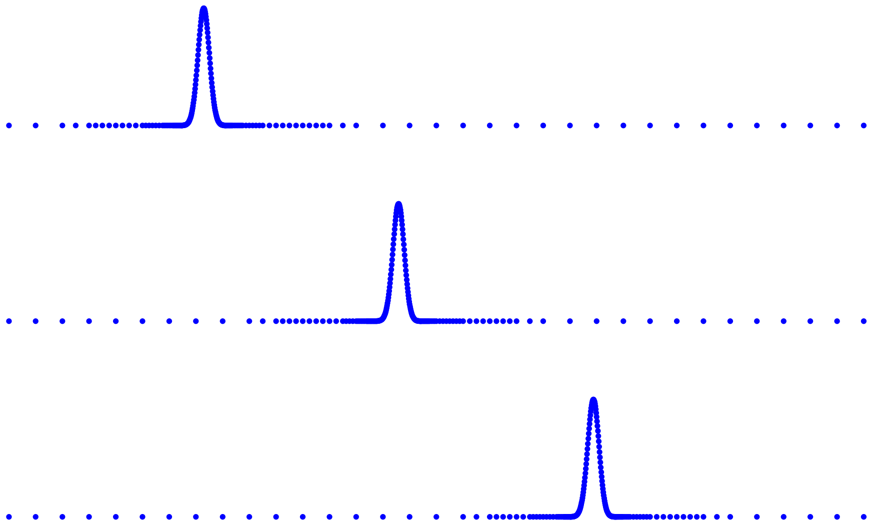


Figure 1: Gaussian peak propagating along x-axis

0.3 Acknowledgements

The work of this thesis has been done under supervision of my advisor Prof. Dr. Andreas Rieder. His patience and kindness during the period when the progress of my work was very slow is greatly appreciated. He carefully reviewed my work and gave me valuable advices step by step which led to the success of this work. And I want to thank Dr. K. R. Hiremath for his help in the knowledge of electrodynamics and computer skills such as g++ coding and Linux system. He also provided some data needed for comparison of simulation results with different methods. Moreover, I want to thank my second advisor Prof. Dr. Christian Wieners, who also carefully reviewed my thesis and gave me some helpful comments.

I also want to thank Wolfgang Müller and Daniel Maurer who supported me in using an eight nodes cluster, ma-otto09, which made the computations in Chapter 5 possible.

Furthermore, I want to thank my former and present colleagues in Research Training Group 1294 for their kindness and friendly fellowships shown to me. Especially, I want to mention Alexander Bulovyatov who helped me with a lot of things such as registration, visa information, health insurance matters and so on when I first arrived in Germany, and Tomas Donald who helped me with latex, and Thomas Gauss to whom I often brought letters written in German for translation, and Kai Sandfort who kindly encouraged me a lot of times when my work had little progress.

Finally, I appreciate the financial support from German Research Foundation(DFG).

Chapter 1

Mathematical modeling of the micro-ring resonator

1.1 Structure of a micro-ring resonator

Analyzing high frequency signal coupling efficiencies of a type of optical waveguide, micro-ring resonator, which is composed of a micro-ring cavity and two straight waveguides is the main purpose of the numerical simulation. The geometry of this micro-ring resonator is described in detail in Figure 1.1. From the position A, the left part of the waveguide WG1 below, a bundle of signal containing continuous frequencies will be launched. The excitation in WG1 is a Gaussian pulse modulating a frequency carrier[1] [35]. Then along with the time evolution we will observe that some parts of the signals of certain frequencies will be switched into the ring cavity and also again be switched into the other straight waveguide, while other parts of the signals will continuously propagate along the WG1 and exit from the right position B of WG1. The numerical simulations of ring resonators have been done using FDTD [16], DGTD [18], [28] and CMT [17], etc. In this paper we will simulate the ring resonator with AWCM and compare the results obtained with FDTD, ISFM and CMT.

1.2 Time domain Maxwell's equations

1.2.1 3D Maxwell's equations

Propagation of electro-magnetic waves is described by Maxwell's Equations, which consist of *Faraday's law*, *Ampere's law*, *Gauss's law for electric field*, and *Gauss's law for magnetic field.* The time dependent Maxwell's Equations in three dimensions in differential form are

[1] This will be explained in detail in Chapter 5.

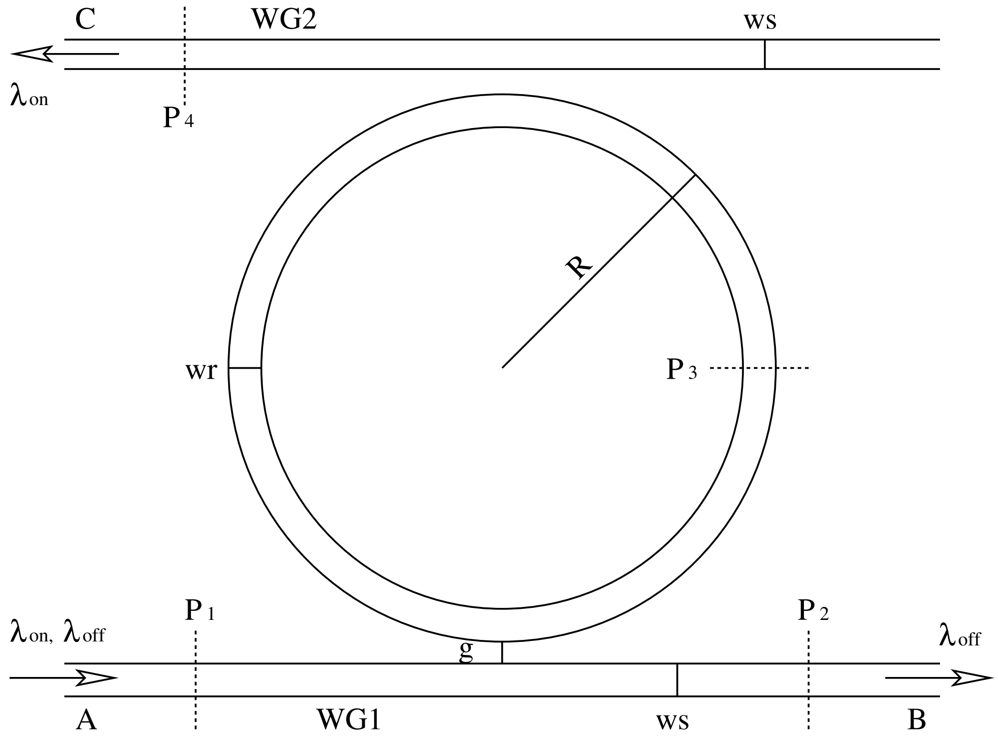


Figure 1.1: A geometric diagram of a micro-ring resonator, which is composed of a circular ring cavity and two lateral straight waveguides. On-resonance and off-resonance signal excited from port A are guided with different directions by the ring resonator. Source: [35].

given by:

$$-\frac{\partial \mathcal{B}}{\partial t} = \nabla \times \mathcal{E} \qquad \text{in} \quad \Omega \times [0, \infty), \tag{1.1a}$$

$$\frac{\partial \mathcal{D}}{\partial t} = \nabla \times \mathcal{H} - \mathcal{J} \qquad \text{in} \quad \Omega \times [0, \infty), \tag{1.1b}$$

$$\nabla \cdot \mathcal{D} = \rho \qquad \text{in} \quad \Omega \times [0, \infty), \tag{1.1c}$$

$$\nabla \cdot \mathcal{B} = 0 \qquad \text{in} \quad \Omega \times [0, \infty), \tag{1.1d}$$

where the symbols in (1.1a) - (1.1d) are:

- $\mathcal{E}$: electric field (volts / meter),
- $\mathcal{D}$: electric flux density (coulombs / meter^2),
- $\mathcal{H}$: magnetic field (amperes / meter),
- $\mathcal{B}$: magnetic flux density (webers / meter^2),
- $\mathcal{J}$: electric current density (amperes / meter^2),

- ρ: free charge density (coulombs / meter^3).

Each of these fields is a three dimensional vector function of four independent variables: x, y, z and t ($(x, y, z) \in \Omega$, $t \in [0, \infty)$), where $\Omega \subset \mathbb{R}^3$ is a bounded domain.

Remark 1.1. 1. Symbols in the time domain equations such as $\mathcal{B}$, $\mathcal{E}$, $\mathcal{D}$, $\mathcal{H}$, and $\mathcal{J}$ are denoted by calligraphic fonts to be distinguished from those in the frequency domain equations. We use bold fonts for the fields in the frequency domain, i.e. **B**, **E**, **D**, **H** and **J**.

2. We will use subindex to denote each component of the vector, for example, $\mathcal{E} = \hat{x}\mathcal{E}_x + \hat{y}\mathcal{E}_y + \hat{z}\mathcal{E}_z$, where $\hat{x}$, $\hat{y}$, $\hat{z}$ are unit vectors along x, y, z respectively. Note that $\mathcal{E}_y$ here does not mean the partial derivative of $\mathcal{E}$ with respect to y.

3. Equations (1.1a), (1.1b) are called *curl equations*.

4. Equations (1.1c), (1.1d) are called *divergence equations*.

5. In linear, isotropic materials, $\mathcal{D}$ is related to $\mathcal{E}$ by a constant called *electrical permittivity*, as well as $\mathcal{B}$ is related to $\mathcal{H}$ by a constant called *magnetic permeability*. These relations are called *constitutive equations*.

$$\mathcal{D} = \varepsilon\mathcal{E} = \varepsilon_0\varepsilon_r\mathcal{E},$$

$$\mathcal{B} = \mu\mathcal{H} = \mu_0\mu_r\mathcal{H},$$

where

- ε: electrical permittivity (farads / meter),
- ε_r: relative permittivity or dielectric constant (dimensionless scalar),
- ε_0: free space permittivity ($8.854187817 \times 10^{-12}$ farads / meter),
- μ: magnetic permeability (henrys / meter),
- μ_r: relative permeability (dimensionless scalar),
- μ_0: free space permeability ($4\pi \times 10^{-7}$ henrys / meter).

For anisotropic materials, the dielectric constant is different for different directions of the electric field, and $\mathcal{D}$ and $\mathcal{E}$ generally have different directions, in this case, the permittivity ε is a matrix:

$$\begin{bmatrix} \mathcal{D}_x \\ \mathcal{D}_y \\ \mathcal{D}_z \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} \begin{bmatrix} \mathcal{E}_x \\ \mathcal{E}_y \\ \mathcal{E}_z \end{bmatrix}. \tag{1.3}$$

In particular, if the off-diagonal entries of the matrix in (1.3) are all zero, we have

$$\begin{bmatrix} \mathcal{D}_x \\ \mathcal{D}_y \\ \mathcal{D}_z \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} & 0 & 0 \\ 0 & \varepsilon_{22} & 0 \\ 0 & 0 & \varepsilon_{33} \end{bmatrix} \begin{bmatrix} \mathcal{E}_x \\ \mathcal{E}_y \\ \mathcal{E}_z \end{bmatrix}. \tag{1.4}$$

This type of medium is called to be *biaxial.* Moreover, if we have $\varepsilon_{11} = \varepsilon_{22}$, the medium is *uniaxial.* In the case of $\varepsilon_{11} = \varepsilon_{22} = \varepsilon_{33}$, it is an *isotropic medium.*

6. In this thesis, we will only deal with the case that there is no free charge density, i.e., $\rho \equiv 0$.

7. Initial conditions and boundary conditions are needed to solve various types of problems.

1.2.2 Decoupling of 2D Maxwell's equations

Assume x, z directions represent the horizontal direction and the vertical direction respectively, and the fields are constant along y-direction[2], thus, the partial derivatives with respect to y vanish in the equations (1.1a) and (1.1b) so that Maxwell's equations are divided into *transverse magnetic mode with respect to y* (TM$_y$) and *transverse electric mode with respect to y* (TE$_y$):

- TM$_y$ mode:

$$\frac{\partial \mathcal{H}_x}{\partial t} = \frac{1}{\mu}\frac{\partial \mathcal{E}_y}{\partial z}, \tag{1.5a}$$

$$\frac{\partial \mathcal{H}_z}{\partial t} = -\frac{1}{\mu}\frac{\partial \mathcal{E}_y}{\partial x}, \tag{1.5b}$$

$$\frac{\partial \mathcal{E}_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial \mathcal{H}_x}{\partial z} - \frac{\partial \mathcal{H}_z}{\partial x} - \mathcal{J}_y\right). \tag{1.5c}$$

- TE$_y$ mode:

$$\frac{\partial \mathcal{E}_x}{\partial t} = \frac{1}{\varepsilon}\left(-\frac{\partial \mathcal{H}_y}{\partial z} - \mathcal{J}_x\right), \tag{1.6a}$$

$$\frac{\partial \mathcal{E}_z}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial \mathcal{H}_y}{\partial x} - \mathcal{J}_z\right), \tag{1.6b}$$

$$\frac{\partial \mathcal{H}_y}{\partial t} = \frac{1}{\mu}\left(\frac{\partial \mathcal{E}_x}{\partial z} - \frac{\partial \mathcal{E}_z}{\partial x}\right). \tag{1.6c}$$

[2] Fields in the 2D Maxwell's equations are still 3D vector fields. However, these are called 2D since the fields do not change along the y direction.

Remark 1.2. 1. TM$_y$ mode and TE$_y$ mode are independent of each other, hence, in the homogeneous case, i.e. $\mathcal{J} = 0$, any solution of 2D Maxwell's equations is a linear combination of solutions of two modes, conversely, any linear combination of solutions of two modes is a solution of 2D Maxwell's equations.

2. In this thesis, we will focus on TM$_y$ mode for our problem of the numerical simulation of the micro-ring resonator.

1.2.3 Reduction to 1D Maxwell's equations

Starting from the 2D Maxwell's equations (1.5), (1.6), we assume further that fields are constant along z direction. Then, the partial derivatives with respect to z vanish in (1.5), (1.6). Thus, TM$_y$ mode and TE$_y$ mode are more simplified as an x-directed, y-polarized *transverse electromagnetic* (TEM) wave and an x-directed, z-polarized *transverse electromagnetic* (TEM)[3] wave, respectively.

- x-directed y-polarized TEM mode:

$$\begin{aligned}\frac{\partial \mathcal{E}_y}{\partial t} &= \frac{1}{\varepsilon}\left(-\frac{\partial \mathcal{H}_z}{\partial x} - \mathcal{J}_y\right),\\ \frac{\partial \mathcal{H}_z}{\partial t} &= -\frac{1}{\mu}\frac{\mathcal{E}_y}{\partial x}.\end{aligned}$$

- x-directed z-polarized TEM mode:

$$\begin{aligned}\frac{\partial \mathcal{E}_z}{\partial t} &= \frac{1}{\varepsilon}\left(\frac{\partial \mathcal{H}_y}{\partial x} - \mathcal{J}_z\right),\\ \frac{\partial \mathcal{H}_y}{\partial t} &= -\frac{1}{\mu}\frac{\mathcal{E}_z}{\partial x}.\end{aligned}$$

Remark 1.3. 1. Similar with 2D case, these two 1D modes are independent of each other.

2. In each of these two modes, if we assume $\mathcal{J} = 0$, by eliminating either electric or magnetic field, we derive traditional 1D wave equation (i.e. $\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}$, where $c^2 = 1/\sqrt{\mu_0\varepsilon_0}$ is the speed of light in vacuum) for magnetic or electric field.

1.3 Numerical methods

Numerical methods to differential equations are indispensable when there is no analytic solution available. The derivatives in differential equations are substituted by numerically approximated ones so that the new equations can be solved with computers. According to

[3] These terminologies here are referenced from those in [35].

the type of numerical discretization, there are various kinds of numerical methods, such as FDTD, ISFM, *finite element method* (FEM) and AWCM, etc. With the guarantee of the certain accuracy analysis, approximate solutions computed from approximated equations are practically useful in application. All these computations are done on a bounded domain Ω.

1.3.1 Incident source

Hard source

A *hard source* is simply specifying $\mathcal{E}$ and $\mathcal{H}$ fields values on some selected points with given time function. For example, in a 1D numerical grid, we can generate a continuous sinusoidal wave of frequency f_0 by hard source for $\mathcal{E}_y$ at position x_{hard}:

$$\mathcal{E}_y|_{x_{hard}}^n = \mathcal{E}_0 \sin(2\pi f_0 n \Delta t),$$

where $\mathcal{E}_0$ is the amplitude of the sinusoidal wave, and n is the index for time stepping. We can also generate another type of hard source, a bandpass Gaussian pulse with zero dc content:

$$\mathcal{E}_y|_{x_{hard}}^n = \mathcal{E}_0 \exp(-[(n-n_0)/n_{decay}]^2) \sin(2\pi f_0 (n-n_0)\Delta t).$$

The pulse is centered at the time-step n_0 and n_{decay} is a scaling factor of the Gaussian amplitude.

An incident source launched by a hard source technique is easy to implement. However, it is not a preferable way of source launching for a long-duration incident wave such as continuous mono-frequency wave. Because when the scattered field propagates back to the hard source points, retro-reflective waves will occur from these locations so that it would contaminate the computation [35]. This problem has been solved by another technique which will be discussed in next subsection.

Total field and scattered field technique

We can excite an arbitrary incident wave using *total field/scattered field* (TF/SF) formulation, see for example, K. R. Umashankar [38] and A. Taflove and S. C. Hagness [35]. Based on the linearity of Maxwell's equations in vacuum, we decompose the electric field and magnetic field as:

$$\mathcal{E}_{\text{total}} = \mathcal{E}_{\text{inc}} + \mathcal{E}_{\text{scat}}, \qquad \mathcal{H}_{\text{total}} = \mathcal{H}_{\text{inc}} + \mathcal{H}_{\text{scat}}.$$

We divide the whole computational domain into two regions (see Figure 1.2). The inside region is total field region, and the outside region is scattered field region. In the total field region the total field is stored in the computer memory, in the scattered field region the scattered field is stored in the computer memory. At the numerical cells near the interface between the total field region and the scattered field region, the numerical derivatives are calculated by the stored variables of different types. We must correct these numerical derivatives at those cells. The incident field values $\mathcal{E}_{inc}$ and $\mathcal{H}_{inc}$ are known beforehand and the total field and scattered field values are unknown. Since the formulation of TF/SF is dependent on each type of numerical method, we will discuss it with AWCM in detail in Chapter 5.

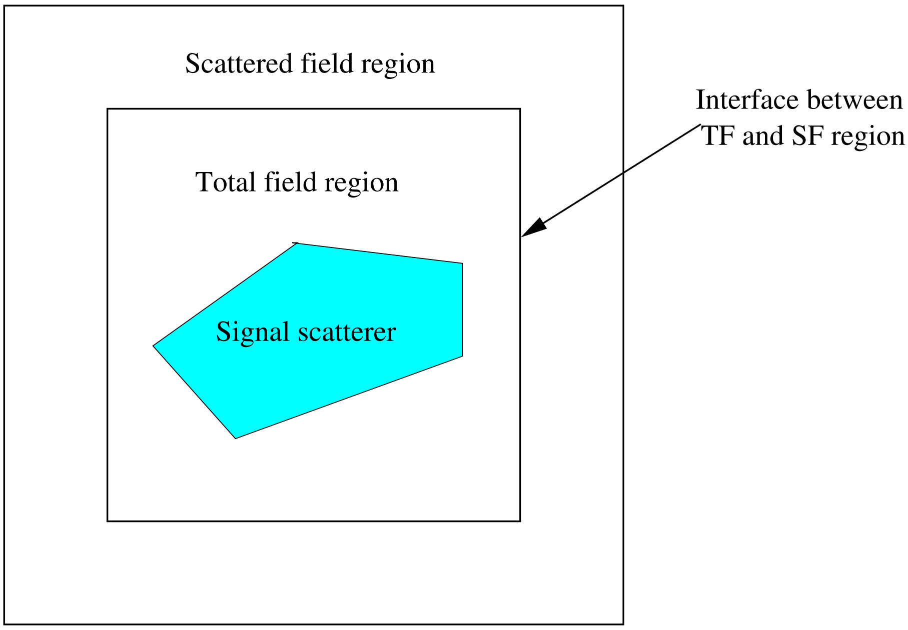


Figure 1.2: Description of the total field and scattered field regions.

1.3.2 Perfectly Matched Layer

In our numerical simulation problems of a micro-ring resonator, we need to simulate unbounded propagations of electromagnetic waves. However, we cannot store infinite number of numerical data in computers or even if we managed to store those data we could not perform numerical calculations on the infinite number of data. Our interests are only in the region where the signals are interacting with the materials. Therefore, we must do simulations on finite, truncated computational domains. When we truncate computational domains, our concern is that those signals scattered by waveguides should disappear from the boundary as if it is exiting from the boundary without any reflection. This is done by adding an absorbing medium around the original computational domain, which absorbs the waves incident into the layer after being scattered by the waveguides in the main domain, [2], [15]. This absorbing medium around the original computational domain is called *perfectly matched layer* (PML).

There are two key points in the theory of PML:

1. the fields match at the interface between the isotropic and anisotropic media, i.e. zero reflection at the interface,

2. after totally transmitted into the PML region, the fields attenuate rapidly in the PML region.

Let us consider a time-harmonic, TE$_y$-polarized plane wave,

$$\mathcal{H}^{inc}(x,z,t) = \Re(\mathbf{H}^{inc}(x,z)\exp(\imath\omega t)),$$

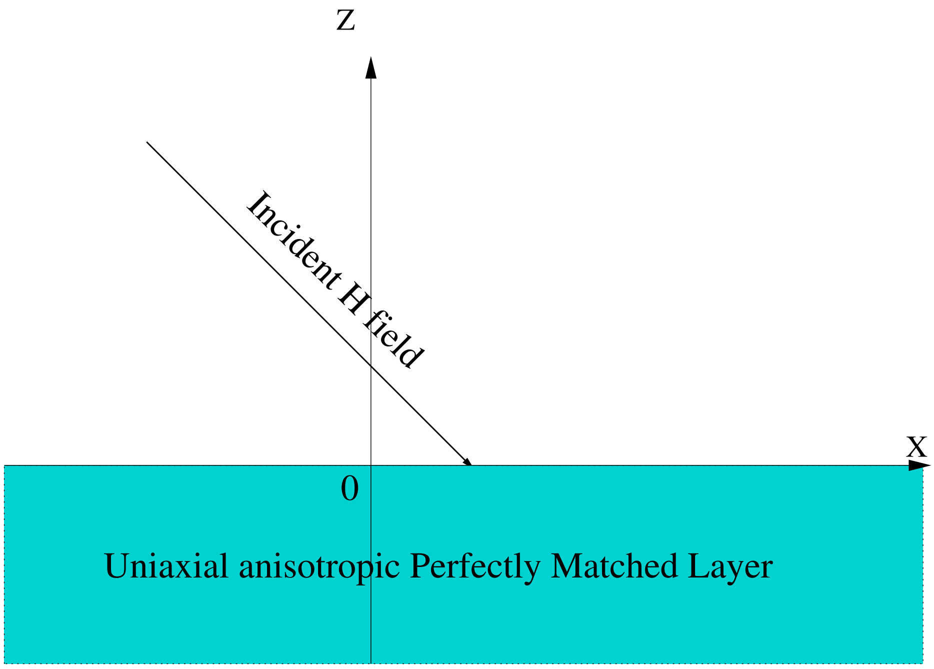


Figure 1.3: A magnetic plane wave incident on the interface between vacuum and PML.

with frequency ω, where

$$\mathbf{H}^{inc}(x,z) = \hat{y} H_0 \exp(-\imath\beta_x^i x - \imath\beta_z^i z)$$

(Figure 1.3), in isotropic space ($z > 0$) is incident on a lossy material ($z < 0$) which is a uniaxial anisotropic medium[4], where $\hat{y}$ is a unit vector along y direction, and H_0 is the amplitude of the sinusoidal wave, $\imath = \sqrt{-1}$, and β_x^i, β_z^i are wavenumbers of the plane wave in the isotropic medium corresponding to x, z direction respectively and the superindex i means *isotropic*. Note that we use the calligraphic fonts for field values in the time domain and the bold fonts for those frequency domain. The interface between two media is the $z = 0$ plane. The fields excited within the uniaxial anisotropic medium satisfy two curls equations (1.1a), (1.1b) with uniaxial constitutive relation.

$$\nabla \times \mathcal{E} = -\mu_0 \mu_r \overline{\overline{\mu}} \frac{\partial \mathcal{H}}{\partial t}, \tag{1.9a}$$

$$\nabla \times \mathcal{H} = \varepsilon_0 \varepsilon_r \overline{\overline{\varepsilon}} \frac{\partial \mathcal{E}}{\partial t}, \tag{1.9b}$$

where ε_r and μ_r are the relative permittivity and permeability of the isotropic space and

$$\overline{\overline{\varepsilon}} = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & b \end{bmatrix}, \qquad \overline{\overline{\mu}} = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & d \end{bmatrix}.$$

[4] See the definition of the *uniaxial medium* in the Remark1.1.

The equations in the system (1.9) in the frequency domain are:

$$\nabla \times \mathbf{E} = -\imath\omega\mu_0\mu_r\overline{\overline{\mu}}\mathbf{H}, \tag{1.10a}$$

$$\nabla \times \mathbf{H} = \imath\omega\varepsilon_0\varepsilon_r\overline{\overline{\varepsilon}}\mathbf{E}. \tag{1.10b}$$

Since derivative of an exponential function is constant times the original function, i.e. $(\exp(ax))' = a\exp(ax)$, it is clear that for any sinusoidal plane wave,

$$\mathbf{A}(x,z) = \hat{y}A_0\exp(-\imath\beta_x x - \imath\beta_z z),$$

the curl operator is equal to a multiplication operator with the vector $-\imath\beta$, i.e.,

$$\nabla\times\ = \beta\times$$

where $\beta = \hat{x}\beta_x + \hat{z}\beta_z$. Note that β is not a 2D vector, indeed, it is a 3D vector whose $\hat{y}$ component is 0.

The incident plane wave with the wavenumber vector β^i after entering the uniaxial anisotropic medium becomes another plane wave with the wavenumber vector $\beta^a = \hat{x}\beta_x^a + \hat{z}\beta_z^a$, where the superindex a means *anisotropic*, thus, (1.10) becomes:

$$\beta^{\mathbf{a}} \times \mathbf{E} = \omega\mu_0\mu_r\overline{\overline{\mu}}\mathbf{H}, \tag{1.11a}$$

$$\beta^{\mathbf{a}} \times \mathbf{H} = -\omega\varepsilon_0\varepsilon_r\overline{\overline{\varepsilon}}\mathbf{E}. \tag{1.11b}$$

Here, if $\overline{\overline{\mu}}$ and $\overline{\overline{\varepsilon}}$ are identity matrices in $\mathbb{R}^{3\times3}$, the equations in (1.10) coincide with the isotropic case, hence, we derive from (1.10a)

$$\mathbf{E}^{inc}(x,z) = (\hat{x}\beta_z^i - \hat{z}\beta_x^i)\cdot H_0/(\omega\varepsilon)\cdot\exp(-\imath\beta_x^i x - \imath\beta_z^i z).$$

It is well known that in the theory of electromagnetic waves, *at the dielectric interface the tangential components of the electric and magnetic field intensities must be continuous.* Now using the continuity, we compute the reflection coefficient Γ of TE$_y$ incident wave at the interface ($z = 0$) of the two half spaces. The reflection coefficient Γ is defined by the ratio of the amplitudes of reflected field to incident field at the interface. In the upper half-space ($z > 0$) where the medium is isotropic, the total field is a superposition of the incident and reflected fields, and the reflected magnetic field

$$\mathbf{H}^{ref} = \hat{y}\Gamma H_0\exp(-\imath\beta_x^i x + \imath\beta_z^i z),$$

thus,

$$\begin{aligned}\mathbf{H}^{up} &= \mathbf{H}^{inc} + \mathbf{H}^{ref}\\ &= \hat{y}H_0(\exp(-\imath\beta_x^i x - \imath\beta_z^i z) + \Gamma\exp(-\imath\beta_x^i x + \imath\beta_z^i z))\\ &= \hat{y}H_0(1 + \Gamma\exp(2\imath\beta_z^i z))\cdot\exp(-\imath\beta_x^i x - \imath\beta_z^i z).\end{aligned}$$

By substituting the $\mathbf{H}^{ref}$ into (1.10a) with the isotropic setting, we get

$$\mathbf{E}^{ref} = (-\hat{x}\beta_z^i - \hat{z}\beta_x^i)\Gamma H_0/(\omega\varepsilon)\exp(-\imath\beta_x^i x + \imath\beta_z^i z),$$

hence,

$$\begin{aligned}
\mathbf{E}^{up} &= \mathbf{E}^{inc} + \mathbf{E}^{ref} \\
&= (\hat{x}\beta_z^i - \hat{z}\beta_x^i)H_0/(\omega\varepsilon)\exp(-\imath\beta_x^i x - \imath\beta_z^i z) + (-\hat{x}\beta_z^i - \hat{z}\beta_x^i)\Gamma H_0/(\omega\varepsilon)\exp(-\imath\beta_x^i x + \imath\beta_z^i z) \\
&= \left[\hat{x}\beta_z^i(1 - \Gamma\exp(2\imath\beta_z^i)) - \hat{z}\beta_x^i(1 + \Gamma\exp(2\imath\beta_z^i))\right] H_0/(\omega\varepsilon)\exp(-\imath\beta_x^i x - \imath\beta_z^i z).
\end{aligned}$$

The wave transmitted into the lower half-space medium which is anisotropic will also be expressed as

$$\begin{aligned}
\mathbf{H}^{low} &= \mathbf{H}^{tra} \\
&= \hat{y}\tau H_0/(\omega\varepsilon)\exp(-\imath\beta_x^a x - \imath\beta_z^a z), \\
\mathbf{E}^{low} &= \mathbf{E}^{tra} \\
&= (\hat{x}\beta_z^a/a - \hat{z}\beta_x^a/b)\tau H_0/(\omega\varepsilon)\exp(-\imath\beta_x^a x - \imath\beta_z^a z),
\end{aligned}$$

where τ is the transmission coefficient, which is defined by the ratio of the amplitudes of the transmitted field and incident field at the interface. Enforcing the continuity of the tangential components of the fields at the interface, we obtain

$$\beta_x^i = \beta_x^a; \quad \Gamma = \frac{\beta_z^i - \beta_z^a/a}{\beta_z^i + \beta_z^a/a}; \quad \tau = 1 + \Gamma = \frac{2\beta_z^i}{\beta_z^i + \beta_z^a/a}.$$

We want to construct an anisotropic medium which perfectly matches (i.e. $\Gamma = 0$), thus, we require $\beta_z^i = \beta_z^a/a$. If we write the equations (1.11) in each components, we have

$$\beta_x^a\mathbf{E}_z - \beta_z^a\mathbf{E}_x = \omega\mu_0\mu c\mathbf{H}_y, \tag{1.12}$$

$$\beta_z^a\mathbf{H}_y = \omega\varepsilon_0\varepsilon_r a\mathbf{E}_x, \tag{1.13}$$

$$\beta_x^a\mathbf{H}_y = -\omega\varepsilon_0\varepsilon_r b\mathbf{E}_z. \tag{1.14}$$

From (1.13), (1.14) we eliminate $\mathbf{E}_x$, $\mathbf{E}_z$ and substitute those into (1.12), then we get the dispersion relation in the uniaxial medium:

$$\beta_x^{a2}/b + \beta_z^{a2}/a = k^2 \cdot c, \tag{1.15}$$

where $k^2 = \omega^2\varepsilon_0\varepsilon_r\mu_0\mu_r$, which satisfies the dispersion relation in the isotropic medium ${\beta_x^i}^2 + {\beta_z^i}^2 = k^2$. Since $\beta_x^a = \beta_x^i$ and $\beta_z^a = a\beta_x^a$, we get from (1.15),

$${\beta_x^i}^2/b + {\beta_z^i}^2 \cdot a = k^2 \cdot c.$$

It is valid if we choose $c = a$ and $b = 1/a$. Hence, the plane wave will be purely transmitted into the uniaxial anisotropic medium if $a = c = 1/b = 1/d$, independent of the angle of incidence. The remaining thing is to find a suitable a such that the transmitted wave attenuates in the medium. This can be done if we choose $a = 1 + \dfrac{\sigma}{\imath\omega\varepsilon_0}$, hence, we have,

$$\overline{\overline{\varepsilon}} = \begin{bmatrix} 1 + \dfrac{\sigma}{\imath\omega\varepsilon_0} & 0 & 0 \\ 0 & 1 + \dfrac{\sigma}{\imath\omega\varepsilon_0} & 0 \\ 0 & 0 & \dfrac{1}{1 + \dfrac{\sigma}{\imath\omega\varepsilon_0}} \end{bmatrix} = \overline{\overline{\mu}}. \tag{1.16}$$

Finally, given a TE$_y$ incident wave, the field intensities in the uniaxial medium are given by

$$\mathbf{H}^{low} = \hat{y}H_0 \cdot \exp(-\imath\beta_x^i x - \imath\beta_z^i z) \cdot \exp(-\alpha_z z),$$

$$\mathbf{E}^{low} = \left[\hat{x}\frac{\beta_z^i}{\omega\varepsilon_0\varepsilon_r} - \hat{z}\frac{\beta_x^i(1 + \dfrac{\sigma}{\imath\omega\varepsilon_0})}{\omega\varepsilon_0\varepsilon_r} \right] \cdot H_0 \cdot \exp(-\imath\beta_x^i x - \imath\beta_z^i z) \cdot \exp(-\alpha_z z),$$

where the attenuating factor is $\alpha_z = \dfrac{\sigma}{\omega\varepsilon_0}\beta_z^i$, using the relation between the wavenumber k and the angular frequency ω, i.e. $v = \dfrac{\omega}{k}$, where $v = \dfrac{1}{\sqrt{\mu\varepsilon}}$ is the speed of the waves, and the definition of impedance $\eta_0 := \sqrt{\dfrac{\mu}{\varepsilon_0}}$, we obtain

$$\alpha_z = \sigma\eta_0\sqrt{\varepsilon_r}\cos\theta^i,$$

where, θ^i is the incident angle of the plane wave (i.e. $\beta_z^i = k\cos\theta^i$). We substitute (1.16) into the matrix form of the equation (1.10), then we have

$$\begin{bmatrix} \dfrac{\partial \mathbf{H}_z}{\partial y} - \dfrac{\partial \mathbf{H}_y}{\partial z} \\ \dfrac{\partial \mathbf{H}_x}{\partial z} - \dfrac{\partial \mathbf{H}_z}{\partial x} \\ \dfrac{\partial \mathbf{H}_y}{\partial x} - \dfrac{\partial \mathbf{H}_x}{\partial y} \end{bmatrix} = \imath\omega\varepsilon_0\varepsilon_r \begin{bmatrix} 1 + \dfrac{\sigma}{\imath\omega\varepsilon_0} & 0 & 0 \\ 0 & 1 + \dfrac{\sigma}{\imath\omega\varepsilon_0} & 0 \\ 0 & 0 & \dfrac{1}{1 + \dfrac{\sigma}{\imath\omega\varepsilon_0}} \end{bmatrix} \begin{bmatrix} \mathbf{E}_x \\ \mathbf{E}_y \\ \mathbf{E}_z \end{bmatrix}. \tag{1.17}$$

We convert these equations from the frequency domain into the time domain with inverse Fourier transformation, then the first two equations in (1.17) in the time domain become

$$\frac{\partial \mathcal{H}_z}{\partial y} - \frac{\partial \mathcal{H}_y}{\partial z} = \varepsilon_0\varepsilon_r\frac{\partial \mathcal{E}_x}{\partial t} + \sigma\varepsilon_r\mathcal{E}_x,$$

$$\frac{\partial \mathcal{H}_x}{\partial z} - \frac{\partial \mathcal{H}_z}{\partial x} = \varepsilon_0\varepsilon_r\frac{\partial \mathcal{E}_y}{\partial t} + \sigma\varepsilon_r\mathcal{E}_y.$$

The inverse Fourier transform of the third equation in (1.17) is not convenient since the term $\imath\omega$ is in the denominator, Gedney [15] introduced a technique which split the inversion into two steps using $\mathbf{D}_z$, that is, first update $\mathbf{D}_z$ using information of the magnetic fields and then update $\mathbf{E}_z$ with the updated $\mathbf{D}_z$:

$$\mathbf{D}_z = \frac{\varepsilon_0\varepsilon_r}{1+\dfrac{\sigma}{\imath\omega\varepsilon_0}}\mathbf{E}_z. \tag{1.18}$$

Then, we have

$$\frac{\partial\mathcal{H}_y}{\partial x} - \frac{\partial\mathcal{H}_x}{\partial y} = \frac{\partial\mathcal{D}_z}{\partial t}.$$

From (1.18), we get

$$\imath\omega\mathbf{D}_z + \frac{\sigma}{\varepsilon_0}\mathbf{D}_z = \imath\omega\varepsilon_0\varepsilon_r\mathbf{E}_z.$$

Again, with the inverse Fourier transform we can derive the relation between $\mathcal{D}_z$ and $\mathcal{E}_z$ in the time domain:

$$\frac{\partial\mathcal{D}_z}{\partial t} + \frac{\sigma}{\varepsilon_0}\mathcal{D}_z = \varepsilon_0\varepsilon_r\frac{\partial\mathcal{E}_z}{\partial t}.$$

Hence, the $\mathcal{E}_z$ field can be updated from $\mathcal{D}_z$ field which has been updated in the previous step. So far, we have considered only for a simple case of a medium which is uniaxial along only one direction. However, in practical, we need to deal with some *corner regions* which are uniaxial along various directions. In these general corner regions, the matrices $\overline{\overline{\varepsilon}}$, $\overline{\overline{\mu}}$, which describe the property of the medium are:

$$\overline{\overline{\varepsilon}} = \overline{\overline{\mu}} = \begin{bmatrix} \dfrac{1}{s_x} & 0 & \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{bmatrix} \begin{bmatrix} s_y & 0 & \\ 0 & \dfrac{1}{s_y} & 0 \\ 0 & 0 & s_y \end{bmatrix} \begin{bmatrix} s_z & 0 & \\ 0 & s_z & 0 \\ 0 & 0 & \dfrac{1}{s_z} \end{bmatrix} = \begin{bmatrix} \dfrac{s_y s_z}{s_x} & 0 & \\ 0 & \dfrac{s_x s_z}{s_y} & 0 \\ 0 & 0 & \dfrac{s_x s_y}{s_z} \end{bmatrix},$$

where $s_x = 1 + \dfrac{\sigma_x}{\imath\omega\varepsilon_0}$, $s_y = 1 + \dfrac{\sigma_y}{\imath\omega\varepsilon_0}$, $s_z = 1 + \dfrac{\sigma_z}{\imath\omega\varepsilon_0}$. Let

$$\widetilde{\mathbf{D}}_z = \varepsilon_0\varepsilon_r\frac{s_x}{s_z}\mathbf{E}_z. \tag{1.19}$$

Note that here we use $\widetilde{\mathbf{D}}_z$ to distinguish it with $\mathbf{D}_z$, which is $s_y\widetilde{\mathbf{D}}_z$. We substitute (1.19) into the second row in the equation of the matrix form (1.17), then,

$$\begin{aligned}\frac{\partial\mathbf{H}_y}{\partial x} - \frac{\partial\mathbf{H}_x}{\partial y} &= \imath\omega s_y\widetilde{\mathbf{D}}_z \\ &= \imath\omega\widetilde{\mathbf{D}}_z + \frac{\sigma_y}{\varepsilon_0}\widetilde{\mathbf{D}}_z.\end{aligned} \tag{1.20}$$

By converting (1.20) into the time domain, we have

$$\frac{\partial\mathcal{H}_y}{\partial x} - \frac{\partial\mathcal{H}_x}{\partial y} = \frac{\partial\widetilde{\mathcal{D}}_z}{\partial t} + \frac{\sigma_y}{\varepsilon_0}\widetilde{\mathcal{D}}_z. \tag{1.21}$$

From equation (1.19) we have

$$\imath\omega\widetilde{\mathbf{D}}_z + \frac{\sigma_z}{\varepsilon_0}\widetilde{\mathbf{D}}_z = \varepsilon_0\varepsilon_r\left(\imath\omega\mathbf{E}_z + \frac{\sigma_x}{\varepsilon_0}\mathbf{E}_z\right).$$

Again with inverse Fourier transform we have the following time domain relation:

$$\frac{\partial\widetilde{\mathcal{D}}_z}{\partial t} + \frac{\sigma_z}{\varepsilon_0}\widetilde{\mathcal{D}}_z = \varepsilon_0\varepsilon_r\left(\frac{\partial\mathcal{E}_z}{\partial t} + \frac{\sigma_x}{\varepsilon_0}\mathcal{E}_z\right).$$

In this way, we are able to incorporate the PML method to our TM$_y$ mode problem (1.5) with $\mathcal{J} = 0$[5]. In this case, the matrices $\overline{\overline{\varepsilon}}$ and $\overline{\overline{\mu}}$ of the medium property in the curl equations in the frequency domain (1.10) are:

$$\overline{\overline{\varepsilon}} = \overline{\overline{\mu}} = \begin{bmatrix} \frac{1}{s_x} & 0 & \\ 0 & s_x & 0 \\ 0 & 0 & s_x \end{bmatrix}\begin{bmatrix} s_z & 0 & \\ 0 & s_z & 0 \\ 0 & 0 & \frac{1}{s_z} \end{bmatrix} = \begin{bmatrix} \frac{s_z}{s_x} & 0 & \\ 0 & s_x s_z & 0 \\ 0 & 0 & \frac{s_x}{s_z} \end{bmatrix}.$$

Using the previous method described above, we get a PML-extended form of TM$_y$ mode equation

$$\frac{\partial\mathcal{B}_x}{\partial t} = \frac{\partial\mathcal{E}_y}{\partial z} \qquad \text{in } \Omega_{\text{PML}} \times [0,\infty), \tag{1.22a}$$

$$\frac{\partial\mathcal{H}_x}{\partial t} + \frac{\sigma_z}{\varepsilon_0}\mathcal{H}_x = \frac{1}{\mu_0}\left(\frac{\partial\mathcal{B}_x}{\partial t} + \frac{\sigma_x}{\varepsilon_0}\mathcal{B}_x\right) \qquad \text{in } \Omega_{\text{PML}} \times [0,\infty), \tag{1.22b}$$

$$\frac{\partial\mathcal{B}_z}{\partial t} = -\frac{\partial\mathcal{E}_y}{\partial x} \qquad \text{in } \Omega_{\text{PML}} \times [0,\infty), \tag{1.22c}$$

$$\frac{\partial\mathcal{H}_z}{\partial t} + \frac{\sigma_x}{\varepsilon_0}\mathcal{H}_z = \frac{1}{\mu_0}\left(\frac{\partial\mathcal{B}_z}{\partial t} + \frac{\sigma_z}{\varepsilon_0}\mathcal{B}_z\right) \qquad \text{in } \Omega_{\text{PML}} \times [0,\infty), \tag{1.22d}$$

$$\frac{\partial\widetilde{\mathcal{D}}_y}{\partial t} + \frac{\sigma_x}{\varepsilon_0}\widetilde{\mathcal{D}}_y = \frac{\partial\mathcal{H}_x}{\partial z} - \frac{\partial\mathcal{H}_z}{\partial x} \qquad \text{in } \Omega_{\text{PML}} \times [0,\infty), \tag{1.22e}$$

$$\frac{\partial\mathcal{E}_y}{\partial t} + \frac{\sigma_z}{\varepsilon_0}\mathcal{E}_y = \frac{1}{\varepsilon}\frac{\partial\widetilde{\mathcal{D}}_y}{\partial t} \qquad \text{in } \Omega_{\text{PML}} \times [0,\infty), \tag{1.22f}$$

where Ω_{PML} is the computational domain extended by PML (See Figure 1.4). Our interest is in the update of $\mathcal{H}_x$, $\mathcal{H}_z$ and $\mathcal{E}_y$, not in the fields such as $\mathcal{B}_x$, $\mathcal{B}_z$ and $\widetilde{\mathcal{D}}_y$. In practical calculation, these auxiliary fields are needed only in the region when the corresponding lossy factor $\sigma_i \neq 0$ $(i = x, z)$. Inside the main computational domain Ω, where all the lossy factors are zero, this system (1.22) coincides with (1.5).

[5] See the Remark 1.4.

Remark 1.4. 1. $\mathscr{J} = 0$, since, in our simulation problem, there is no electric current density source.

2. Initial conditions: The initial conditions for these fields are dependent on each type of problems we want to solve. For example, in the case we test the propagation of a Gaussian pulse in the waveguides, the electric field $\mathcal{E}_y$ will be given as a Gaussian pulse, and all other fields are zeros initially. When we need to continuously input some type of waveguide modes, the initial conditions of all the fields are zeros, except that the electric field values on some particular positions need to be corrected at each time step either by the hard source or the total field and scattered field source.

$$\begin{aligned} \mathcal{B}_x(x,z,0) &= {\mathcal{B}_x}^{ini}(x,z), & \mathcal{H}_x(x,z,0) &= {\mathcal{H}_x}^{ini}(x,z), \\ \mathcal{B}_z(x,z,0) &= {\mathcal{B}_z}^{ini}(x,z), & \mathcal{H}_z(x,z,0) &= {\mathcal{H}_z}^{ini}(x,z), \\ \widetilde{\mathcal{D}}_y(x,z,0) &= \widetilde{\mathcal{D}}_y^{ini}(x,z), & \mathcal{E}_y(x,z,0) &= {\mathcal{E}_y}^{ini}(x,z), \end{aligned}$$

for $\forall\ (x,z) \in \Omega_{PML}$, where ${\mathcal{B}_x}^{ini}(x,z)$, ${\mathcal{H}_x}^{ini}(x,z)$, ${\mathcal{B}_z}^{ini}(x,z)$, ${\mathcal{H}_z}^{ini}(x,z)$, $\widetilde{\mathcal{D}}_y^{ini}(x,z)$ and ${\mathcal{E}_y}^{ini}(x,z)$ are given functions.

3. Boundary conditions: The boundary conditions of all fields (i.e. on the outermost boundary) are always zeros. Here the assumption is that the propagating fields start to attenuate exponentially after entering PML region and become almost zeros when they arrive at the outermost boundary.

$$\begin{aligned} \mathcal{B}_x(x,z,t) &= 0, & \mathcal{H}_x(x,z,t) &= 0, \\ \mathcal{B}_z(x,z,t) &= 0, & \mathcal{H}_z(x,z,t) &= 0, \\ \widetilde{\mathcal{D}}_y(x,z,t) &= 0, & \mathcal{E}_y(x,z,t) &= 0, \end{aligned}$$

for $\forall\ (x,z) \in \partial\Omega_{PML}$, $\forall\ t \in [0,\infty)$, where $\partial\Omega_{\text{PML}}$ is the boundary of Ω_{PML}.

1.4 Numerical approximation of derivatives

Let us consider a scalar function $u(x,z,t)$ such that $u(x,z,t) \in \mathbf{C}^2(\Omega_{\text{PML}}) \times [0,\infty)$, where Ω_{PML} is a 2D rectangular domain, i.e. $\Omega_{\text{PML}} = [x_{left},\ x_{right}] \times [z_{bottom},\ z_{top}]$. We uniformly divide Ω_{PML} into $N_x \times N_z$ sub domains. Let $\Delta x = \dfrac{x_{right} - x_{left}}{N_x}$, $\Delta z = \dfrac{z_{top} - z_{bottom}}{N_z}$. Denote $u|_{i,j}^{n} = u(i\Delta x,\ j\Delta z,\ n\Delta t)$, where Δt is the time step, $i,\ j,\ n \in \dfrac{1}{2}\mathbb{Z} = \left\{\dfrac{1}{2}m : m \in \mathbb{Z}\right\}$.

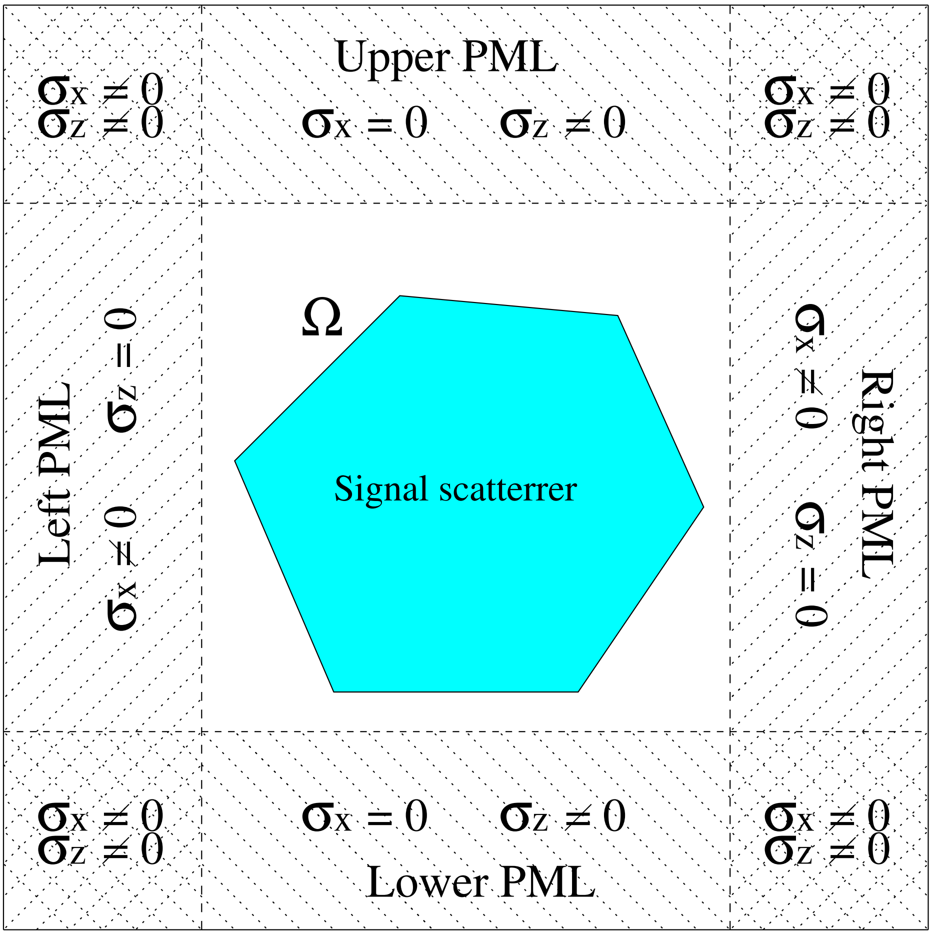


Figure 1.4: Description of a computational domain for the simulation of general time domain Maxwell's equations: computational domain is surrounded by PML region. The inside white domain is the original computational domain Ω. The whole domain including PML is Ω_{PML}.

Approximation of time derivatives

Consider the approximation of the time derivative of $u(x, z, t)$ at the grid point $(i\Delta x, j\Delta z)$, at the time step $n\Delta t$. From the Taylor series expansion, we have:

$$u|_{i,j}^{n+1/2} = u|_{i,j}^{n} + \frac{\Delta t}{2}\frac{\partial u}{\partial t}(i\Delta x,\, j\Delta z,\, n\Delta t) + \frac{\Delta t^2}{8}\frac{\partial^2 u}{\partial t^2}(i\Delta x,\, j\Delta z,\, n\Delta t) + O(\Delta t^3), \tag{1.25a}$$

$$u|_{i,j}^{n-1/2} = u|_{i,j}^{n} - \frac{\Delta t}{2}\frac{\partial u}{\partial t}(i\Delta x,\, j\Delta z,\, n\Delta t) + \frac{\Delta t^2}{8}\frac{\partial^2 u}{\partial t^2}(i\Delta x,\, j\Delta z,\, n\Delta t) + O(\Delta t^3). \tag{1.25b}$$

By subtracting (1.25b) from (1.25a), we get

$$\frac{\partial u}{\partial t}(i\Delta x,\, j\Delta z,\, n\Delta t) = \frac{u|_{i,j}^{n+1/2} - u|_{i,j}^{n-1/2}}{\Delta t} + O(\Delta t^2).$$

This is a central finite difference scheme (or symmetric scheme) which has second order accuracy. We use leap-frog time stepping[6] for our TM_y modes system extended with PML:

[6] This is a scheme whose electric field and magnetic field are half time step staggered.

(1.22). First, if we consider semi-numerical scheme, i.e. only time derivatives are approximated, we have the following.

$$\mathcal{B}_x|^{n+1/2} = \mathcal{B}_x|^{n-1/2} + \Delta t \, \frac{\partial \mathcal{E}_y{}^n}{\partial z}, \tag{1.26a}$$

$$\mathcal{H}_x|^{n+1/2} = \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \, \mathcal{H}_x|^{n-1/2} + \frac{1}{\mu_0} \left(\frac{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \, \mathcal{B}_x|^{n+1/2} - \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \, \mathcal{B}_x|^{n-1/2} \right), \tag{1.26b}$$

$$\mathcal{B}_z|^{n+1/2} = \mathcal{B}_z|^{n-1/2} - \Delta t \, \frac{\partial \mathcal{E}_y|^n}{\partial x}, \tag{1.26c}$$

$$\mathcal{H}_z|^{n+1/2} = \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \, \mathcal{H}_z|^{n-1/2} + \frac{1}{\mu_0} \left(\frac{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \, \mathcal{B}_z|^{n+1/2} - \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \, \mathcal{B}_z|^{n-1/2} \right), \tag{1.26d}$$

$$\widetilde{\mathcal{D}}_y|^{n+1} = \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \, \widetilde{\mathcal{D}}_y|^n + \frac{\Delta t}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \left(\frac{\partial \mathcal{H}_x|^{n+1/2}}{\partial z} - \frac{\partial \mathcal{H}_z|^{n+1/2}}{\partial x} \right), \tag{1.26e}$$

$$\mathcal{E}_y|^{n+1} = \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \, \mathcal{E}_y|^n + \frac{1}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \, \frac{1}{\varepsilon} \left(\widetilde{\mathcal{D}}_y|^{n+1} - \widetilde{\mathcal{D}}_y|^n \right), \tag{1.26f}$$

where fields are only discretized for the time variable. The super index represents the corresponding time step.

Approximation of spatial derivatives

We have established a mathematical model, system (1.22), for the numerical simulations of micro-ring resonators. Now according to the methods of approximating the spatial derivatives, there are *finite difference time domain* (FDTD), *interpolating scaling function method* (ISFM), *adaptive wavelet collocation method* (AWCM), etc. In the next chapter, we will discuss FDTD.

Chapter 2

Finite difference time domain method

The simplest way of approximating a derivative of a given function $f(x) \in \mathbf{C}^3(a,b)$ at one point $x_0 \in (a,b)$ is finite difference. The derivative $f'(x_0)$ which is the slope of the tangential line at x_0 is approximated with the slope of another secant line which passes two points near that point (See Figure 2.1). This approximation is called finite difference. Finite difference can be obtained by truncating the Taylor series of the function at that point.

- forward difference scheme with x_0 and a forward point $x_0 + h$:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + O(h), \tag{2.1a}$$

- backward difference scheme with x_0 and a backward point $x_0 - h$:

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} + O(h), \tag{2.1b}$$

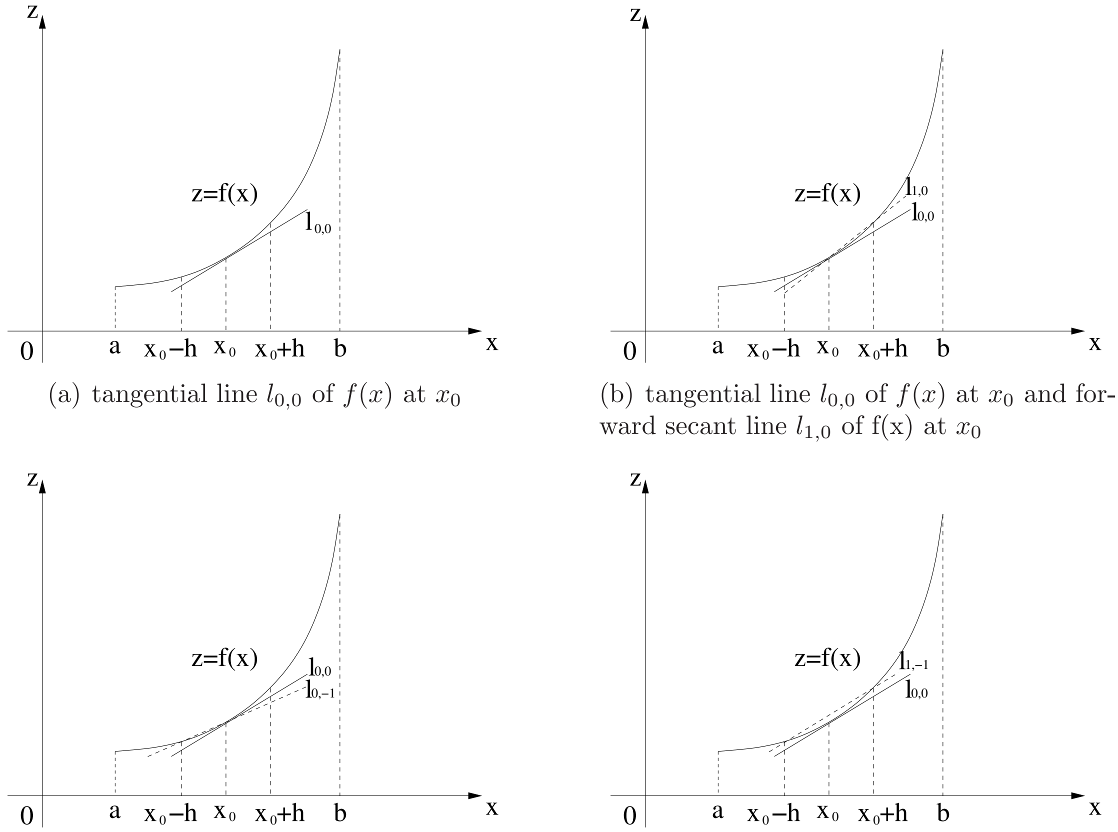
- central difference scheme with a forward point $x_0 + h$ and a backward point $x_0 - h$:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} + O(h^2). \tag{2.1c}$$

Hence, forward and backward finite differences have first order accuracies, while central difference has second order accuracy. When we solve Maxwell's equations, normally we use central difference, which is symmetric. According to the arrangement of numerical grid points of different components of electric field and magnetic field, there are several numerical schemes, such as *the staggered uncollocated scheme*, *the unstaggered collocated scheme* and *the staggered collocated scheme*, [35].

2.1 Yee's scheme

The staggered uncollocated scheme is called Yee's scheme [43] (Figure: 2.2 and 2.3). In Yee's scheme, not only different fields are not collocated on the same positions, but also each different component of the same fields is not collocated on the same grid point. Furthermore,



(a) tangential line $l_{0,0}$ of $f(x)$ at x_0

(b) tangential line $l_{0,0}$ of $f(x)$ at x_0 and forward secant line $l_{1,0}$ of f(x) at x_0

(c) tangential line $l_{0,0}$ of $f(x)$ at x_0 and backward secant line $l_{0,-1}$ of f(x) at x_0

(d) tangential line $l_{0,0}$ of $f(x)$ at x_0 and central secant line $l_{1,-1}$ of f(x) at x_0

Figure 2.1: Lines involved with exact and approximate derivatives of $f(x)$ at x_0

not every grid point in the Yee's lattice is assigned to a component of a field. In detail, there is no field component on points $(i\Delta x, j\Delta y, k\Delta z)$, where i,j,$k \in \mathbb{Z}$. There is one grid point for electric field between every two successive points for magnetic field of the same type, and there is one grid point for magnetic field between every two successive points for electric field of the same type. Thus, each field is updated using the spatial derivatives of other field on both side of it. The finite difference scheme is therefore a central scheme which has second order of accuracy.

Let us consider the semi-numerical equations system (1.26) discretized on the two-dimensional Yee's grid (Figure 2.3). We consider the spatial derivatives in (1.26), such as $\dfrac{\partial \mathcal{E}_y^n}{\partial x}$, $\dfrac{\partial \mathcal{E}_y|^n}{\partial z}$, $\dfrac{\partial \mathcal{H}_x|^{n+1/2}}{\partial z}$, $\dfrac{\partial \mathcal{H}_z|^{n+1/2}}{\partial x}$. Denote $u|_{i,j}^n = u(i\Delta x,\, j\Delta z,\, n\Delta t)$, where $u \in \{\mathcal{E}_y,\, \mathcal{H}_z,\, \mathcal{H}_x\}$, for $i,\, j,\, n \in \dfrac{1}{2}\mathbb{Z}$.

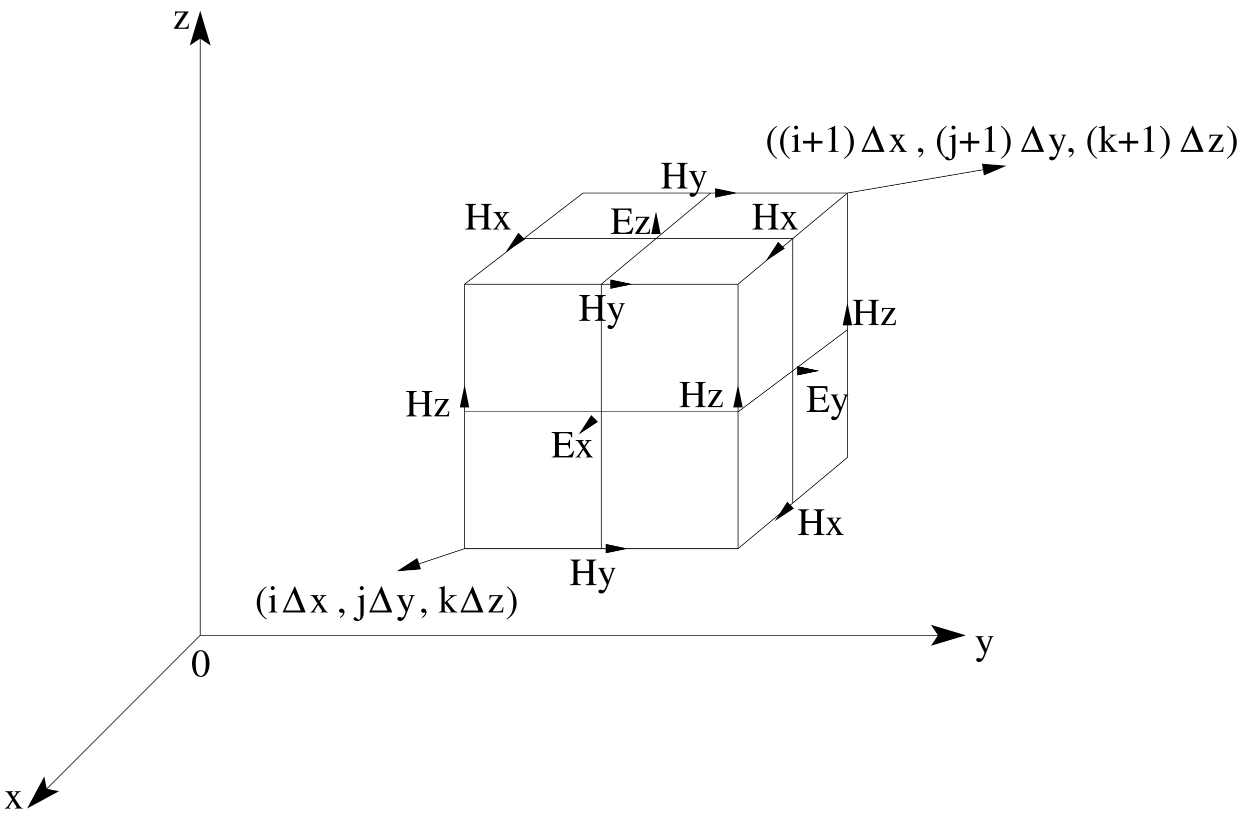


Figure 2.2: Yee's grids in three dimensional space: uncollocated staggered grid. Source: A. Taflove, Susan C. Hagness, *Computational electrodynamics, the finite time difference time domain method, third edition, 2005*, pp. 59.

$$\frac{\partial \mathcal{E}_y|^n_{i,j+1/2}}{\partial z} = \frac{\mathcal{E}_y|^n_{i,\ j+1} - \mathcal{E}_y|^n_{i,j}}{\Delta z} + O(\Delta z^2) \tag{2.2a}$$

$$\frac{\partial \mathcal{E}_y|^n_{i+1/2,\,j}}{\partial x} = \frac{\mathcal{E}_y|^n_{i+1,\,j} - \mathcal{E}_y|^n_{i,\,j}}{\Delta x} + O(\Delta x^2), \tag{2.2b}$$

$$\frac{\partial \mathcal{H}_x|^{n+1/2}_{i,\,j}}{\partial z} = \frac{\mathcal{H}_x|^{n+1/2}_{i,\,j+1/2} - \mathcal{H}_x|^{n+1/2}_{i,\,j-1/2}}{\Delta z} + O(\Delta z^2), \tag{2.2c}$$

$$\frac{\partial \mathcal{H}_z|^{n+1/2}_{i,\,j}}{\partial x} = \frac{\mathcal{H}_z|^{n+1/2}_{i+1/2,\,j} - \mathcal{H}_z|^{n+1/2}_{i-1/2,\,j}}{\Delta x} + O(\Delta x^2). \tag{2.2d}$$

We obtain the approximation of the Maxwell's equations system extended with PML by substituting these finite difference approximations into the semi-numerical system (1.26).

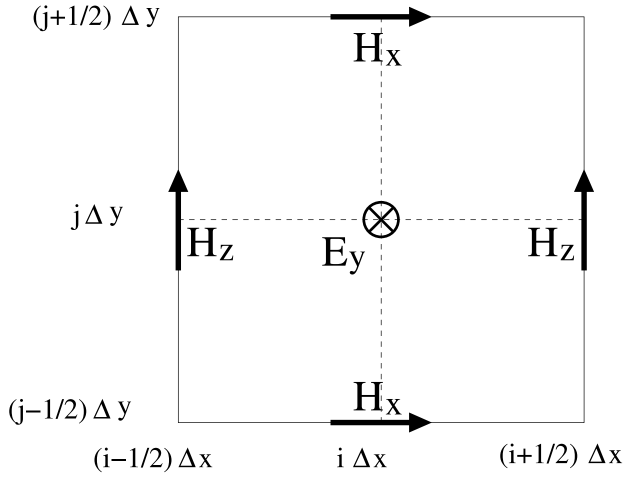


Figure 2.3: Yee's grids in two dimensional space (for TM$_y$ mode): uncollocated staggered grid

$$\mathcal{B}_x|_{i,\,j+1/2}^{n+1/2} = \mathcal{B}_x|_{i,\,j+1/2}^{n-1/2} + \frac{\Delta t}{\Delta z}(\mathcal{E}_y|_{i,\,j+1}^{n} - \mathcal{E}_y|_{i,\,j}^{n}), \tag{2.3a}$$

$$\mathcal{H}_x|_{i,\,j+1/2}^{n+1/2} = \frac{1-\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{H}_x|_{i,\,j+1/2}^{n-1/2} + \frac{1}{\mu_0}\left(\frac{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{B}_x|_{i,\,j+1/2}^{n+1/2} - \frac{1-\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{B}_x|_{i,\,j+1/2}^{n-1/2}\right), \tag{2.3b}$$

$$\mathcal{B}_z|_{i+1/2,\,j}^{n+1/2} = \mathcal{B}_z|_{i+1/2,\,j}^{n-1/2} - \frac{\Delta t}{\Delta x}(\mathcal{E}_y|_{i+1,\,j}^{n} - \mathcal{E}_y|_{i,\,j}^{n}), \tag{2.3c}$$

$$\mathcal{H}_z|_{i+1/2,\,j}^{n+1/2} = \frac{1-\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}\mathcal{H}_z|_{i+1/2,\,j}^{n-1/2} + \frac{1}{\mu_0}\left(\frac{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}\mathcal{B}_z|_{i+1/2,\,j}^{n+1/2} - \frac{1-\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}\mathcal{B}_z|_{i+1/2,\,j}^{n-1/2}\right), \tag{2.3d}$$

$$\widetilde{\mathcal{D}}_y|_{i,\,j}^{n+1} = \frac{1-\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}\widetilde{\mathcal{D}}_y|_{i,\,j}^{n} + \frac{\Delta t}{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}} \cdot \left(\frac{\mathcal{H}_x|_{i,\,j+1/2}^{n+1/2} - \mathcal{H}_x|_{i,\,j-1/2}^{n+1/2}}{\Delta z} - \frac{\mathcal{H}_z|_{i+1/2,\,j}^{n+1/2} - \mathcal{H}_z|_{i-1/2,\,j}^{n+1/2}}{\Delta x}\right), \tag{2.3e}$$

$$\mathcal{E}_y|_{i,\,j}^{n+1} = \frac{1-\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{E}_y|_{i,\,j}^{n} + \frac{1}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\frac{1}{\varepsilon}\left(\widetilde{\mathcal{D}}_y|_{i,\,j}^{n+1} - \widetilde{\mathcal{D}}_y|_{i,\,j}^{n}\right). \tag{2.3f}$$

Remark 2.1. 1. Initial conditions for the difference system (2.3):

$$\mathcal{B}_x|_{i,\,j+1/2}^{-1/2} = \mathcal{B}_x{}^{ini}(i\Delta x, (j+1/2)\Delta z), \quad \mathcal{H}_x|_{i,\,j+1/2}^{-1/2} = \mathcal{H}_x{}^{ini}(i\Delta x, (j+1/2)\Delta z),$$

$$\mathcal{B}_z|_{i+1/2,\,j}^{-1/2} = \mathcal{B}_z{}^{ini}((i+1/2)\Delta x, j\Delta z), \quad \mathcal{H}_z|_{i+1/2,\,j}^{-1/2} = \mathcal{H}_z{}^{ini}((i+1/2)\Delta x, j\Delta z),$$

$$\widetilde{\mathcal{D}}_y|_{i,\,j}^{0} = \widetilde{\mathcal{D}}_y^{ini}(i\Delta x, j\Delta z), \qquad \mathcal{E}_y|_{i,\,j}^{0} = \mathcal{E}_y{}^{ini}(i\Delta x, j\Delta z),$$

for all $i,\, j \in \mathbb{Z}$ such that $i \in [0, N_x]$, $j \in [0, N_z]$.

2. Boundary conditions for the difference system (2.3):

$$\mathcal{B}_x|_{i,\,j+1/2}^{n-1/2} = 0,\ \mathcal{H}_x|_{i,\,j+1/2}^{n-1/2} = 0 : \forall (i,j) \in \mathbb{Z}^2 \text{ such that } i \in \{0, N_x\} \text{ or } j \in \{0, N_z - 1\};$$

$$\mathcal{B}_z|_{i+1/2,\,j}^{n-1/2} = 0,\ \mathcal{H}_z|_{i+1/2,\,j}^{n-1/2} = 0 : \forall (i,j) \in \mathbb{Z}^2 \text{ such that } i \in \{0, N_x - 1\} \text{ or } j \in \{0, N_z\};$$

$$\widetilde{\mathcal{D}}_y|_{i,\,j}^{n} = 0,\ \mathcal{E}_y|_{i,\,j}^{n} = 0 : \qquad \forall (i,j) \in \mathbb{Z}^2 \text{ such that } i \in \{0, N_x\} \text{ or } j \in \{0, N_z\};$$

where $n \in \mathbb{N}_0$.

2.2 Unstaggered collocated scheme

Unlike Yee's scheme, the unstaggered collocated scheme uses the same grid point for all the components of the both of the electric and magnetic fields (Figure: 2.4). There is no "empty point" (i.e. point which is assigned to no component of the either field, like $(i+1/2)\Delta x,\ (j+1/2)\Delta z)$, where $i,\, j \in \mathbb{Z}$ in Figure 2.3).

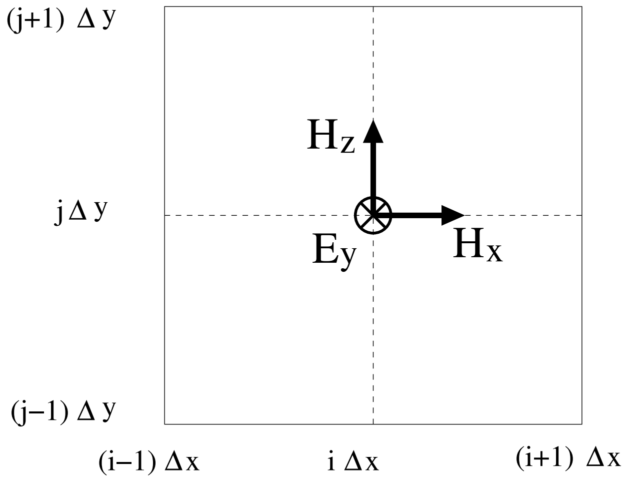


Figure 2.4: Unstaggered collocated scheme in two dimensional space (for TM$_y$ mode)

Then, the spatial derivatives in (1.26) are approximated as the following:

$$\frac{\partial \mathcal{E}_y|_{i,j}^{n}}{\partial z} = \frac{\mathcal{E}_y|_{i,j+1}^{n} - \mathcal{E}_y|_{i,j-1}^{n}}{2\Delta z} + O(\Delta z^2), \tag{2.6a}$$

$$\frac{\partial \mathcal{E}_y|_{i,j}^{n}}{\partial x} = \frac{\mathcal{E}_y|_{i+1,j}^{n} - \mathcal{E}_y|_{i-1,j}^{n}}{2\Delta x} + O(\Delta x^2), \tag{2.6b}$$

$$\frac{\partial \mathcal{H}_x|_{i,j}^{n+1/2}}{\partial z} = \frac{\mathcal{H}_x|_{i,j+1}^{n+1/2} - \mathcal{H}_x|_{i,j-1}^{n+1/2}}{2\Delta z} + O(\Delta z^2), \tag{2.6c}$$

$$\frac{\partial \mathcal{H}_z|_{i,j}^{n+1/2}}{\partial x} = \frac{\mathcal{H}_z|_{i+1,j}^{n+1/2} - \mathcal{H}_z|_{i-1,j}^{n+1/2}}{2\Delta x} + O(\Delta x^2). \tag{2.6d}$$

If we substitute (2.6) into the semi-numerical system (1.26), we obtain:

$$\mathcal{B}_x|_{i,j}^{n+1/2} = \mathcal{B}_x|_{i,j}^{n-1/2} + \frac{\Delta t}{2\Delta z} \cdot (\mathcal{E}_y|_{i,j+1}^{n} - \mathcal{E}_y|_{i,j-1}^{n}), \tag{2.7a}$$

$$\mathcal{H}_x|_{i,j}^{n+1/2} = \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{H}_x|_{i,j}^{n-1/2} + \frac{1}{\mu_0} \left(\frac{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{B}_x|_{i,j}^{n+1/2} - \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{B}_x|_{i,j}^{n-1/2} \right), \tag{2.7b}$$

$$\mathcal{B}_z|_{i,j}^{n+1/2} = \mathcal{B}_z|_{i,j}^{n-1/2} - \frac{\Delta t}{2\Delta x} (\mathcal{E}_y|_{i+1,j}^{n} - \mathcal{E}_y|_{i-1,j}^{n}), \tag{2.7c}$$

$$\mathcal{H}_z|_{i,j}^{n+1/2} = \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \mathcal{H}_z|_{i,j}^{n-1/2} + \frac{1}{\mu_0} \left(\frac{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \mathcal{B}_z|_{i,j}^{n+1/2} - \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \mathcal{B}_z|_{i,j}^{n-1/2} \right), \tag{2.7d}$$

$$\begin{aligned} \widetilde{\mathcal{D}}_y|_{i,j}^{n+1} = & \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \widetilde{\mathcal{D}}_y|_{i,j}^{n} + \frac{\Delta t}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \\ & \left(\frac{\mathcal{H}_x|_{i,j+1}^{n+1/2} - \mathcal{H}_x|_{i,j-1}^{n+1/2}}{2\Delta z} - \frac{\mathcal{H}_z|_{i+1,j}^{n+1/2} - \mathcal{H}_z|_{i-1,j}^{n+1/2}}{2\Delta x} \right), \end{aligned} \tag{2.7e}$$

$$\mathcal{E}_y|_{i,j}^{n+1} = \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{E}_y|_{i,j}^{n} + \frac{1}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \frac{1}{\varepsilon} \left(\widetilde{\mathcal{D}}_y|_{i,j}^{n+1} - \widetilde{\mathcal{D}}_y|_{i,j}^{n} \right). \tag{2.7f}$$

Remark 2.2. 1. Initial conditions for the difference system (2.7):

$$\begin{aligned}
\mathcal{B}_x|_{i,j}^{-1/2} &= \mathcal{B}_x{}^{ini}(i\Delta x, j\Delta z), & \mathcal{H}_x|_{i,j}^{-1/2} &= \mathcal{H}_x{}^{ini}(i\Delta x, j\Delta z),\\
\mathcal{B}_z|_{i,j}^{-1/2} &= \mathcal{B}_z{}^{ini}(i\Delta x, j\Delta z), & \mathcal{H}_z|_{i,j}^{-1/2} &= \mathcal{H}_z{}^{ini}(i\Delta x, j\Delta z),\\
\widetilde{\mathcal{D}}_y|_{i,j}^{0} &= \widetilde{\mathcal{D}}_y^{ini}(i\Delta x, j\Delta z), & \mathcal{E}_y|_{i,j}^{0} &= \mathcal{E}_y{}^{ini}(i\Delta x, j\Delta z),
\end{aligned}$$

for all i, $j \in \mathbb{Z}$ such that $i \in [0, N_x]$, $j \in [0, N_z]$.

2. Boundary conditions for the difference system (2.7):

$$\begin{aligned}
\mathcal{B}_x|_{i,j}^{n-1/2} &= 0, & \mathcal{H}_x|_{i,j}^{n-1/2} &= 0,\\
\mathcal{B}_z|_{i,j}^{n-1/2} &= 0, & \mathcal{H}_z|_{i,j}^{n-1/2} &= 0,\\
\widetilde{\mathcal{D}}_y|_{i,j}^{n} &= 0, & \mathcal{E}_y|_{i,j}^{n} &= 0,
\end{aligned}$$

for all i and j such that either $i \in \{0, N_x\}$ or $j \in \{0, N_z\}$, and for $n \in \mathbb{N}_0$.

3. Order of approximation:
The local error of the difference schemes (2.3), (2.7) are both $O(\Delta t^2) + O(\Delta x^2) + O(\Delta z^2)$. The orders of accuracy for the spatial derivatives approximations discussed above, i.e. (2.2), (2.6) are all two. We can increase the order of the spatial derivatives by increasing the number of points on each side of the corresponding grid point used in approximating derivatives. These schemes are called *higher order finite difference scheme.*

From the (2.6), we notice that the unstaggered collocated scheme is essentially a combination of four staggered independent Yee's schemes (Figure: 2.5), that is, these four parts are unaffected by each other during the implementation. We can easily know that, similarly, in 1D or 3D case we can also decompose the unstaggered collocated scheme into several independent parts, and the number of independent parts is decided by the level of the dimension. This behavior is more clearly explained when we use a 1-point source in the scheme. If the initial conditions of all other points except the source point are zeros, then, only the independent part that contains the source point is being updated during time stepping while other parts stay zeros. Δx in Yee's scheme (Figure 2.3) plays the same role as $2\Delta x$ in the unstaggered scheme (Figure 2.4). Therefore, double number of discretization is needed for the unstaggered scheme in order to get the same accuracy.

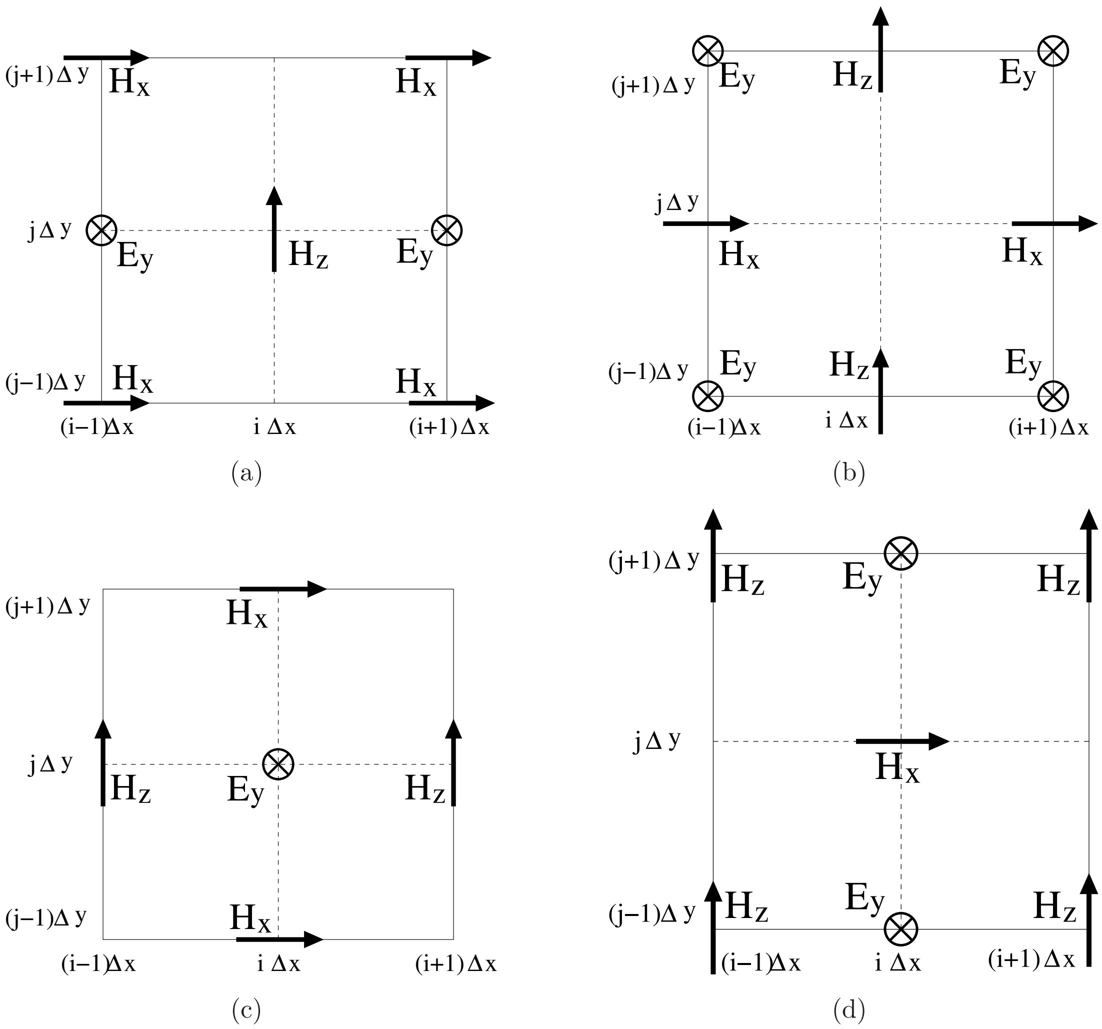


Figure 2.5: Decomposition of the 2D unstaggered collocated scheme (Figure 2.4) into four staggered independent Yee's schemes

2.3 Numerical dispersion and stability

Let us consider TM$_y$ mode equation (1.5) in homogeneous medium with the assumption $\mathcal{J} = 0$[1]:

$$\frac{\partial \mathcal{H}_x}{\partial t} = \frac{1}{\mu}\frac{\partial \mathcal{E}_y}{\partial z}, \tag{2.10a}$$

$$\frac{\partial \mathcal{H}_z}{\partial t} = -\frac{1}{\mu}\frac{\partial \mathcal{E}_y}{\partial x}, \tag{2.10b}$$

$$\frac{\partial \mathcal{E}_y}{\partial t} = \frac{1}{\varepsilon}\left(\frac{\partial \mathcal{H}_x}{\partial z} - \frac{\partial \mathcal{H}_z}{\partial x}\right). \tag{2.10c}$$

[1] Throughout the whole thesis we will use this assumption that there is no current density source.

When we solve Maxwell's equations numerically, it is inevitable for us to discuss the behavior of numerical dispersion and stability, which also influence our choice for the spatial discretization size and time step. We consider the difference system (2.3) in the main computational domain Ω (Figure 1.3) only, where all the lossy factors, σ's, are zero and no auxiliary variable such as $\mathcal{B}$ or $\widetilde{\mathcal{D}}$ is needed, then the system simplifies into:

$$\mathcal{H}_x|_{i,j+1/2}^{n+1/2} = \mathcal{H}_x|_{i,j+1/2}^{n-1/2} + \frac{\Delta t}{\mu\Delta z}(\mathcal{E}_y|_{i,j+1}^{n} - \mathcal{E}_y|_{i,j}^{n}), \tag{2.11a}$$

$$\mathcal{H}_z|_{i+1/2,j}^{n+1/2} = \mathcal{H}_z|_{i+1/2,j}^{n-1/2} - \frac{\Delta t}{\mu\Delta x}(\mathcal{E}_y|_{i+1,j}^{n} - \mathcal{E}_y|_{i,j}^{n}), \tag{2.11b}$$

$$\mathcal{E}_y|_{i,j}^{n+1} = \mathcal{E}_y|_{i,j}^{n} + \frac{\Delta t}{\varepsilon}\left(\frac{\mathcal{H}_x|_{i,j+1/2}^{n+1/2} - \mathcal{H}_x|_{i,j-1/2}^{n+1/2}}{\Delta z} - \frac{\mathcal{H}_z|_{i+1/2,j}^{n+1/2} - \mathcal{H}_z|_{i-1/2,j}^{n+1/2}}{\Delta x}\right). \tag{2.11c}$$

2.3.1 Numerical dispersion

Lemma 2.3. *If a pair of real numbers k_x and k_z satisfies*

$$\omega^2\mu\varepsilon = k_x^2 + k_z^2, \tag{2.12}$$

then, the following set (2.13) of the fields is a solution to the system (2.10),

$$\mathcal{H}_x(x,z,t) = -\frac{k_z}{\mu\omega}\exp(\imath(\omega\, t - k_x\, x - k_z\, z)), \tag{2.13a}$$

$$\mathcal{H}_z(x,z,t) = \frac{k_x}{\mu\omega}\exp(\imath(\omega\, t - k_x\, x - k_z\, z)), \tag{2.13b}$$

$$\mathcal{E}_y(x,z,t) = \exp(\imath(\omega\, t - k_x\, x - k_z\, z)), \tag{2.13c}$$

conversely, if there exists a pair of real numbers k_x and k_z such that (2.13) is a solution to the system (2.10), then k_x and k_z yield (2.12).

Proof. If k_x and k_z satisfy (2.12), we can easily check that (2.13) is a solution of (2.10) simply by substituting (2.13) into (2.10).

Conversely, if there exist k_x and k_z such that (2.13) is a solution to the system (2.10), we can obtain (2.12) by substituting (2.13) into (2.10) and simplifying it. □

The solution (2.13) is a plane sinusoidal wave of angular frequency ω. Suppose that k_x, k_z are components of a wavevector $\vec{k}$, i.e. , $\vec{k} = k_x\hat{x} + k_z\hat{z}$, then from (2.12) we have the analytic relation between phase velocity v and wavevector $\vec{k}$,

$$v = \frac{\omega}{|\vec{k}|}, \tag{2.14}$$

where phase velocity $v = \dfrac{1}{\sqrt{\mu\varepsilon}}$ and $|\vec{k}| = \sqrt{k_x^2 + k_z^2}$.

Optical waves of various frequencies travel at the same speed v in homogeneous medium. However, it is not the same for the numerical case, that is, the solution to the numerical difference system (2.11) is different from that to (2.13) in wavenumbers, for given fixed angular frequency ω. This causes the phase velocity $\tilde{v}$ of the numerical wave differs from the analytic velocity v. And the numerical phase velocity $\tilde{v}$ depends on the angular frequecy ω, number of space discretization, ratio of smallest time step size and space mesh size and propagation direction. This phenomena is called *numerical dispersion*. The choice of the smallest spatial discretization size is restricted due to the analysis of numerical dispersion of the FDTD. In the numerical dispersion analysis, our aim is to calculate $\tilde{v}$ of the corresponding numerical plane sinusoidal wave with frequency ω and compare $\tilde{v}$ with v.

Lemma 2.4. *If a pair of real numbers $\tilde{k}_x$ and $\tilde{k}_z$ satisfies*

$$\left[\frac{1}{v\Delta t}\sin\left(\frac{\omega\Delta t}{2}\right)\right]^2 = \left[\frac{1}{\Delta x}\sin\left(\frac{\tilde{k}_x\Delta x}{2}\right)\right]^2 + \left[\frac{1}{\Delta z}\sin\left(\frac{\tilde{k}_z\Delta z}{2}\right)\right]^2, \tag{2.15}$$

where $v = 1/\sqrt{\mu\varepsilon}$, then the following set (2.16) is a solution to the difference system (2.11),

$$\mathcal{H}_x|^n_{I,\,J+1/2} = -\frac{\Delta t}{\mu\Delta z}\frac{\sin\left(\tilde{k}_z\Delta z/2\right)}{\sin\left(\omega\Delta t/2\right)}\exp(\imath(\omega\, n\Delta t - \tilde{k}_x I\Delta x - \tilde{k}_z(J+1/2)\Delta z)), \tag{2.16a}$$

$$\mathcal{H}_z|^n_{I+1/2,\,J} = \frac{\Delta t}{\mu\Delta x}\frac{\sin\left(\tilde{k}_x\Delta x/2\right)}{\sin\left(\omega\Delta t/2\right)}\exp(\imath(\omega\, n\Delta t - \tilde{k}_x(I+1/2)\Delta x - \tilde{k}_z J\Delta z)), \tag{2.16b}$$

$$\mathcal{E}_y|^n_{I,\,J} = \exp(\imath(\omega\, n\Delta t - \tilde{k}_x I\Delta x - \tilde{k}_z J\Delta z)). \tag{2.16c}$$

Conversely, if there exists a pair of real numbers $\tilde{k}_x$ and $\tilde{k}_z$ such that (2.16) is a solution to the difference system (2.11), then $\tilde{k}_x$ and $\tilde{k}_z$ yield (2.15).

Proof. The proof of this lemma is essentially the same as that of Lemma 2.3. □

The equation (2.15) is *numerical dispersion relation* of the FDTD with Yee's scheme [35], from which we will analyze the relation between $\tilde{v}$ and v. The numerical phase velocity $\tilde{v}$ is defined by:

$$\tilde{v} := \frac{\omega}{\left|\vec{\tilde{k}}\right|}, \tag{2.17}$$

where $\vec{\tilde{k}} = \tilde{k}_x\hat{x} + \tilde{k}_z\hat{z}$. We shall only consider the case of $\Delta x = \Delta z \equiv \Delta$, moreover, we define a term called *CFL stability factor*[2]:

$$S := v\Delta t/\Delta, \tag{2.18}$$

and we know that wavelength λ is related to the angular frequency ω by

$$\lambda\omega = 2\pi v, \tag{2.19}$$

[2] The term is named after Richard Courant, Kurt Friedrichs, and Hans Lewy who described it in their 1928 paper [6].

Now we define the *grid sampling density* $N_\lambda := \lambda/\Delta$. From (2.18) and (2.19) we have $\Delta t = S\Delta/v$ and $\lambda = 2\pi v/\omega$, respectively, hence, the dispersion relation (2.15) becomes

$$\frac{1}{S^2}\sin^2\left(\frac{\pi S}{N_\lambda}\right) = \sin^2\left(\frac{1}{2}\left|\vec{\tilde{k}}\right|\Delta\cos\phi\right) + \sin^2\left(\frac{1}{2}\left|\vec{\tilde{k}}\right|\Delta\sin\phi\right), \tag{2.20}$$

where $\phi = \arctan(\tilde{k}_z/\tilde{k}_x)$, which is the propagation angle of the wave with respect to x direction. And we use $\tilde{v}_\phi$ to denote the numerical phase velocity with respect to the propagation angle ϕ. The dispersion coefficients are different along different propagation directions. Here we will only discuss two directions which are relatively simple: $\phi = 0$ and $\phi = \pi/4$. First, for the case that $\phi = 0$, the equation (2.20) simplifies into:

$$\frac{1}{S}\sin\left(\frac{\pi S}{N_\lambda}\right) = \sin\left(\frac{1}{2}\left|\vec{\tilde{k}}\right|\Delta\right).$$

Thus we get the corresponding numerical phase velocity:

$$\tilde{v}_0 = \frac{\omega}{\left|\vec{\tilde{k}}\right|} = \gamma_0 v,$$

where the *dispersion coefficient* γ_0[3] is

$$\gamma_0 = \frac{\pi}{N_\lambda \arcsin\left[\frac{1}{S}\sin\left(\frac{\pi S}{N_\lambda}\right)\right]}.$$

From this we see that the dispersion coefficient γ_0 is dependent on the choice of both the stability factor S and the grid sampling density N_λ. Next, we come to the case that $\phi = \pi/4$. In this case, the equation (2.20) becomes

$$\frac{1}{S}\sin\left(\frac{\pi S}{N_\lambda}\right) = \sqrt{2}\sin\left(\frac{1}{2}\left|\vec{\tilde{k}}\right|\Delta\right).$$

Thus we get the corresponding numerical phase velocity:

$$\tilde{v}_{\pi/4} = \frac{\omega}{\left|\vec{\tilde{k}}\right|} = \gamma_{\pi/4} v,$$

where the dispersion coefficient $\gamma_{\pi/4}$ is

$$\gamma_{\pi/4} = \frac{\pi}{\sqrt{2}N_\lambda \arcsin\left[\frac{1}{\sqrt{2}S}\sin\left(\frac{\pi S}{N_\lambda}\right)\right]}.$$

As N_λ increases, the coefficient increases towards 1 (See Figure 2.6, 2.7). In Figure 2.7, we tested with CFL factors $1/\sqrt{2}$ times those of Figure 2.6. We need more numbers of numerical meshes in one wavelength to get more accurate numerical phase velocities. In the simulation of ring-resonator with FDTD [16], the smallest mesh size is $13.6nm$, which guarantees at least 100 cells inside a wavelength around $1.5\mu m$. In this case, the error of the numerical dispersion is less than 0.001, see Figure 2.6, 2.7.

[3] γ_ϕ is the dispersion coefficient with respect to the propagation angle ϕ.

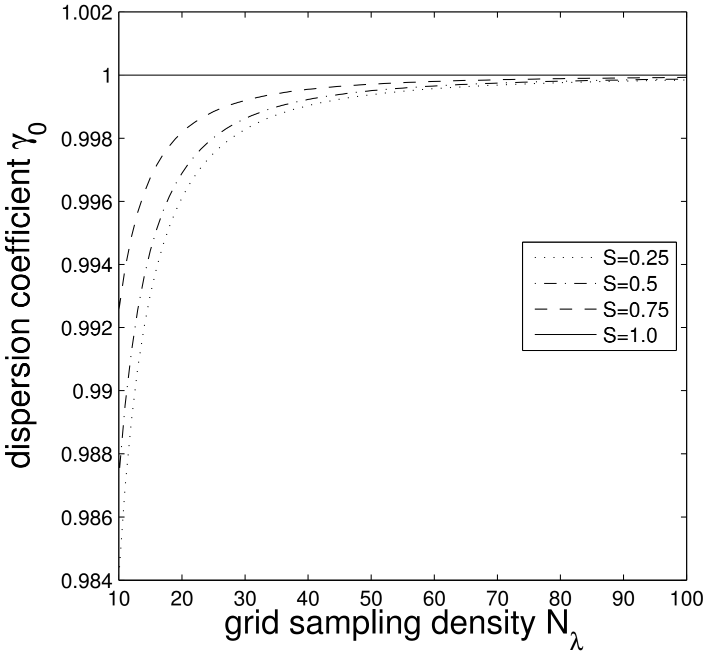


Figure 2.6: Dependencies of dispersion coefficient γ_0 on the grid sampling density N_λ with different fixed CFL factors. Note that when $S = 1$, i.e., there is no numerical dispersion along the propagation angle $\phi = 0$, see the magic time step in [35].

2.3.2 Numerical stability

We cannot choose freely the smallest time step Δt regardless of the mesh size Δx, Δz. The choice of Δt is restricted by the choice of spatial mesh size: Δx, Δz. Otherwise, we have numerical instability, which blows up the numerical data after several time steps. The principal idea of the numerical stability analysis is that the amplification factor of time difference operator is less than that of the curl operator. We will derive the stability condition for the Yee's FDTD scheme for the three dimension case[4]. We rewrite the Maxwell's curl's equations in a homogeneous medium,

$$\frac{\partial \mathcal{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \mathcal{H}, \tag{2.21a}$$

$$\frac{\partial \mathcal{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \mathcal{E}. \tag{2.21b}$$

Let

$$\mathcal{E} = \sqrt{\mu/\varepsilon}\widetilde{\mathcal{E}},$$
$$\mathcal{V} = \mathcal{H} + j\widetilde{\mathcal{E}},$$

[4] These ideas are mainly from [34, 35].

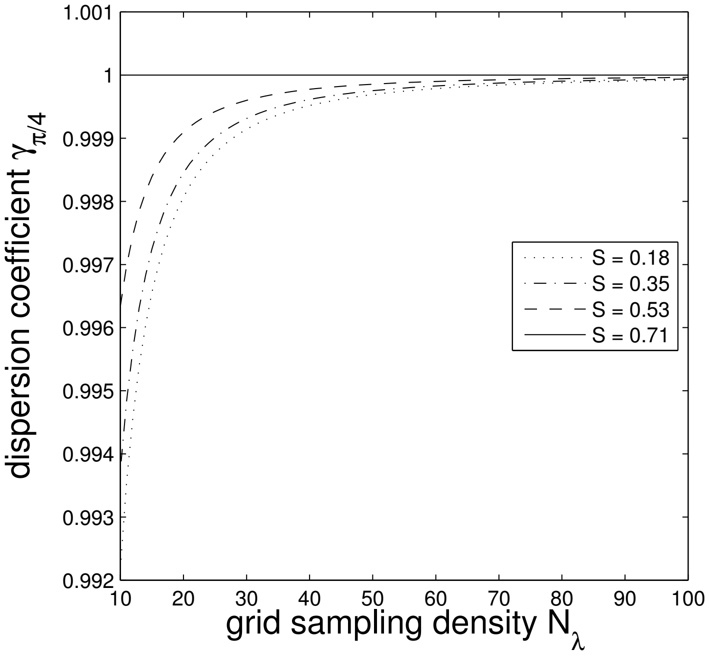


Figure 2.7: Dependencies of dispersion coefficient $\gamma_{\pi/4}$ on the grid sampling density N_λ with different fixed CFL factors. Like the case $\phi = 0$, there is no numerical dispersion when the CFL factor $S = 1/\sqrt{2}$.

where $j = \sqrt{-1}$, then we can combine the two curl equations (2.21) into one equation:

$$\frac{\partial \mathcal{V}}{\partial t} = \frac{j}{\sqrt{\mu\varepsilon}} \nabla \times \mathcal{V}. \tag{2.22}$$

Then the Yee's FDTD scheme of the equation (2.22) becomes:

$$\frac{\mathcal{V}|^{n+1/2}_{p,q,r} - \mathcal{V}|^{n-1/2}_{p,q,r}}{\Delta t} = \frac{j}{\sqrt{\mu\varepsilon}} \widetilde{\nabla} \times \mathcal{V}|^{n}_{p,q,r}, \tag{2.23}$$

where $\widetilde{\nabla}$ is "discretized gradient" with respect to central finite difference of second order with Yee's scheme, i.e., whose three components are central finite difference approximations of partial derivatives with respect to corresponding directions. We use von Neumann method or Fourier method[5] to analyze the numerical stability. We set

$$\mathcal{V}|^{n}_{p,q,r} = V_0 \alpha^n \exp(\imath(k_x p\Delta x + k_y q\Delta y + k_z r\Delta z)), \tag{2.24}$$

where V_0 is a constant 3D vector and α is amplification factor. Our task is to derive the condition which guarantees the numerical stability of the finite difference scheme, i.e., $|\alpha| \leq 1$. We substitute (2.24) into (2.23) to obtain:

$$\frac{\alpha^{1/2} - \alpha^{-1/2}}{\Delta t} V_0 = -\frac{2}{\sqrt{\mu\varepsilon}} \left(\frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2}, \frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2}, \frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2} \right) \times V_0. \tag{2.25}$$

[5] See for example [21].

If we consider V_0 as a column vector and write the equation (2.25) in matrix form, then

$$\frac{\alpha^{1/2}-\alpha^{-1/2}}{\Delta t}V_0 = \frac{2}{\sqrt{\mu\varepsilon}}AV_0, \tag{2.26}$$

where

$$A = \begin{bmatrix} 0 & -\frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2} & \frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2} \\ \frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2} & 0 & -\frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2} \\ -\frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2} & \frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2} & 0 \end{bmatrix}.$$

Thus, from (2.26), we know that $\dfrac{\sqrt{\mu\varepsilon}(\alpha^{1/2}-\alpha^{-1/2})}{2\Delta t}$ is an eigenvalue of the matrix A. By solving

$$|sI - A| = 0,$$

where I is the 3D identity matrix, we get,

$$s\left(s^2 + \frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2} + \frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2} + \frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2}\right) = 0.$$

Since $\alpha^{1/2}-\alpha^{-1/2}$ cannot be zero, we have

$$\left(\frac{\sqrt{\mu\varepsilon}(\alpha^{1/2}-\alpha^{-1/2})}{2\Delta t}\right)^2 + \frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2} + \frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2} + \frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2} = 0. \tag{2.27}$$

After simplification, we have

$$\alpha^2 - (2-2\eta)\alpha + 1 = 0, \tag{2.28}$$

where

$$\eta = 2v\Delta t\sqrt{\frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2} + \frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2} + \frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2}}.$$

Note that $v = 1/\sqrt{\varepsilon\mu}$, thus by solving (2.28), we obtain

$$\alpha = 1-\eta \pm \sqrt{(1-\eta)^2-1}.$$

It is easy to see that $|\alpha| \le 1$ if and only if

$$0 \le \eta \le 2.$$

Hence,

$$v\Delta t\sqrt{\frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2} + \frac{\sin^2(k_y\Delta y/2)}{(\Delta y)^2} + \frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2}} \le 1. \tag{2.29}$$

We require that the inequality (2.29) should be held for all possible k_x, k_y and k_z, thus we have the stability condition of the Yee's FDTD scheme:

$$\Delta t \leq \frac{1}{v\sqrt{\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}+\frac{1}{(\Delta z)^2}}}.$$

Remark 2.5. 1. 2D TM$_y$ mode can be understood as a special case of 3D case, where

$$\mathcal{H}=\hat{x}\mathcal{H}_x+\hat{z}\mathcal{H}_z \quad \text{and} \quad \mathcal{E}=\hat{y}\mathcal{E}_y.$$

Thus,

$$\mathcal{V}=\mathcal{H}+j\mathcal{E}=\hat{x}\mathcal{H}_x+j\hat{y}\mathcal{E}_y+\hat{z}\mathcal{H}_z.$$

We set

$$\mathcal{V}|^n_{p,r}=V_0\alpha^n\exp(\imath(k_xp\Delta x+k_zr\Delta z)),$$

where V_0 is a constant 3D vector, then instead of (2.25), we have

$$\frac{\alpha^{1/2}-\alpha^{-1/2}}{\Delta t}V_0=-\frac{2}{\sqrt{\mu\varepsilon}}\left(\frac{\sin^2(k_x\Delta x/2)}{(\Delta x)^2},0,\frac{\sin^2(k_z\Delta z/2)}{(\Delta z)^2}\right)\times V_0, \tag{2.30}$$

and the remaining steps of the stability analysis is the same as that of 3D case, thus we have,

$$\Delta t \leq \frac{1}{v\sqrt{\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta z)^2}}}.$$

2. For 1D TEM$_y$ mode the process of the stability proof is similar. The stability condition for 1D case is:

$$\Delta t \leq \frac{1}{v\sqrt{\frac{1}{(\Delta x)^2}}}=\frac{\Delta x}{v}.$$

We define $S=\frac{v\Delta t}{\Delta x}$, which is (CFL) stability factor in one dimensional case. Then the stability condition is $S \leq 1$.

3. For the stability of the Uncollocated staggered scheme, which is a combination of several independent Yee's schemes, we apply the stability criterion for each of the independent Yee's scheme whose spatial mesh size is double of that of the original scheme itself. Then we have the stability condition for the Uncollocated Staggered scheme in which the upper bound for Δt is two times that of Yee's scheme.

$$\Delta t \leq \frac{2}{v\sqrt{\frac{1}{(\Delta x)^2}+\frac{1}{(\Delta y)^2}+\frac{1}{(\Delta z)^2}}}.$$

Chapter 3

Interpolating scaling functions method

The Taylor series method is not the only way of deriving the central finite difference scheme (2.1c). We can also obtain the scheme using the concept of *Lagrangian interpolation* (see [37]), which can be straightforwardly extended to other numerical schemes by replacing the Lagrangian polynomials with other types of functions.

We know that there is a unique polynomial of degree less than n that interpolates n distinct given points. We can explicitly represent this unique polynomial in the following form.

Definition 3.1. *To any set of n distinct real data points,*

$$\{(x_j, y_j) \in \mathbb{R}^2 \,|\, j = 1, 2, \cdots, n \text{ and } x_m \neq x_p \text{ for } m \neq p\},$$

the Lagrangian interpolating polynomial is defined by

$$P(x) := \sum_{k=1}^{n} y_k \ell_k(x), \tag{3.1}$$

where the Lagrangian basis polynomial

$$\ell_j(x) = \prod_{\substack{1 \leq k \leq n \\ k \neq j}} \frac{x - x_k}{x_j - x_k}.$$

The error of the Lagrangian interpolation is stated in the following theorem.

Theorem 3.2. *Let x_1, x_2, $\cdots$, x_n be n distinct real numbers in $[a, b]$, and $g \in C^n[a, b]$. Then for $x \in [a, b]$ there exists $\xi(x)$ in (a, b) with*

$$g(x) = P(x) + \frac{g^{(n)}(\xi(x))}{n!}(x - x_1)(x - x_2) \cdots (x - x_n),$$

where P is the Lagrangian interpolating polynomial with n points $(x_j, g(x_j))_{1 \leq j \leq n}$.

Proof. See, for example, K. E. Atkinson [1]. □

Now we come to the derivation of central differences with Lagrangian interpolation. We approximate a function $f \in C^3(a,b)$ with the Lagrangian interpolating polynomial $P(x)$ at three points, $(x_j, f(x_j))$ for $j = 1, 2, 3$, where

$$a < x_1 = x_0 - h, \qquad x_2 = x_0, \qquad x_3 = x_0 + h < b.$$

Then, (3.1) becomes

$$P(x) = f(x_0 - h)\ell_1(x) + f(x_0)\ell_2(x) + f(x_0 + h)\ell_3(x), \tag{3.2}$$

where the Lagrangian basis polynomials ℓ_1, ℓ_2 and ℓ_3 of the three points $x_0 - h$, x_0 and $x_0 + h$ are as following:

$$\ell_1(x) = \frac{x - x_0}{(x_0 - h) - x_0} \frac{x - (x_0 + h)}{(x_0 - h) - (x_0 + h)}, \tag{3.3a}$$

$$\ell_2(x) = \frac{x - (x_0 - h)}{x_0 - (x_0 - h)} \frac{x - (x_0 + h)}{x_0 - (x_0 + h)}, \tag{3.3b}$$

$$\ell_3(x) = \frac{x - (x_0 - h)}{(x_0 + h) - (x_0 - h)} \frac{x - x_0}{(x_0 + h) - x_0}. \tag{3.3c}$$

Differentiating on both sides of (3.2) and substituting $x = x_0$ into it, we obtain the central difference scheme:

$$f'(x_0) \approx P'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h}.$$

Since 3-points Lagrangian interpolation is exact for the polynomial of degree less than 3, the approximation (3.2) above has second order accuracy.

We have shown that finite difference can be obtained by differentiating the local approximation of a function with Lagrangian interpolations. Now we consider replacing these local polynomial functions with another type of functions called *interpolating scaling functions*[1] (ISF's). In this thesis we will call this method *interpolating scaling function method* (ISFM). There are several different ways of constructing ISF's. We can construct ISF's by *iterative interpolation processes*, which does not involve the concept of wavelets, see G. Deslauriers and S. Dubuc (1989) [10]. N. Satio and G. Beylkin (1992) have shown that ISF is an autocorrelation of Daubechies compactly supported scaling function [31]. W. Sweldens [32] proved that one can also obtain ISF by *lifting* a set of biorthogonal wavelet filter called *Lazy wavelet*. Like FDTD in previous chapter, we also consider uniform meshes, that is, only scaling functions of the same level of resolution are involved.

[1] These functions were sometimes also named as *interpolating wavelets*, or *Deslauries-Dubuc interpolating functions*, or *fundamental interpolating functions*, see for example D. L. Donoho [12], M. Fujii [14] and S. Dubuc [13].

3.1 Wavelets

3.1.1 Multi-resolution approximations

Our purpose is to decompose functions in $L^2(\mathbb{R})$ according to different *resolution levels*[2]. We start with the mathematical definition of *multi-resolution approximations* (MRA) introduced by Mallat [25, 24] and Meyer [27]. First, we need the definition of the *Riesz basis*.

Definition 3.3 (Riesz basis)**.** *Assume that V is a subspace of $L^2(\mathbb{R})$. We call a set of functions*

$$\{e_n \in V \mid n \in \mathbb{Z}\},$$

a Riesz basis of V if there exist $A > 0$ and B such that any function f in V can be uniquely decomposed into

$$f(\cdot) = \sum_{n=-\infty}^{+\infty} a_n e_n(\cdot),$$

where $a_n \in \mathbb{R}$ and satisfies the following inequality

$$A \left\| f \right\|_{L^2}^2 \leq \sum_{-\infty}^{+\infty} |a_n|^2 \leq B \left\| f \right\|_{L^2}^2 .$$

The notation $\| \cdot \|_{L^2}$ refers to $L^2(\mathbb{R})$ norm, i.e.,

$$\|f\|_{L^2} = \left(\int |f(x)|^2 dx \right)^{1/2}, \quad f \in L^2(\mathbb{R}).$$

Now, we come to the definition of the MRA.

Definition 3.4 (MRA)**.** *A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ is a multiresolution approximation if the following 6 properties are satisfied:*

$$\forall\ j, k \in \mathbb{Z}, f(\cdot) \in V_j \Leftrightarrow f\left(\cdot - \frac{k}{2^j}\right) \in V_j,$$

$$\forall\ j \in \mathbb{Z}, V_j \subset V_{j+1},$$

$$\forall\ j \in \mathbb{Z}, f(\cdot) \in V_j \Leftrightarrow f(2\cdot) \in V_{j+1},$$

$$\lim_{j\to-\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\},$$

$$\lim_{j\to+\infty} V_j = \overline{\left(\bigcup_{j=-\infty}^{+\infty} V_j \right)} = L^2(R).$$

There exists $\phi(\cdot) \in L^2(\mathbb{R})$ such that $\{\phi(\cdot - k)\}_{k\in\mathbb{Z}}$ is a Riesz basis of V_0.

[2] This term will be defined immediately.

Remark 3.5. 1. The subindex j is the *resolution level.* Translating or shifting does not change the resolution level of a function, while dilating or contraction does. The higher the level j, functions with the more detailed information are contained in V_j.

2. For an $L^2(\mathbb{R})$ function f, we define its orthogonal projection $P_{V_j}f$ into the subspace V_j of $L^2(\mathbb{R})$. We know that

$$\lim_{j\to-\infty}\|P_{V_j}f\|_{L^2}=0 \text{ and } \lim_{j\to+\infty}\|f-P_{V_j}\|_{L^2}=0.$$

The first equation means that, if a function loses every detail of it, then nothing is left. And the second equation means that a function recovers its original information by obtaining all details of every resolution level.

3. We can easily check that, for any $j\in\mathbb{Z}$, $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$, which is defined by

$$\phi_{j,k}(\cdot):=2^{j/2}\phi(2^j\cdot-k),$$

is a Riesz basis of V_j. In particular, $\{\phi_{1,k}\}_{k\in\mathbb{Z}}$ is a Riesz basis of V_1. Since $V_0\subset V_1$, we represent ϕ as the linear combination of the basis $\{\phi_{1,k}\}_{k\in\mathbb{Z}}$ of V_1:

$$\phi(\cdot)=\sum_{k=-\infty}^{+\infty}h_k\phi_{1,k},$$

or

$$\phi(\cdot)=\sqrt{2}\sum_{k=-\infty}^{+\infty}h_k\phi(2\cdot-k), \tag{3.4}$$

where $h_k\in\mathbb{R}$. We call ϕ *scaling function*, and the equation (3.4) *scaling equation.* The sequence of coefficients $\{h_k\}_{k\in\mathbb{Z}}$ is called *the filter of scaling function* ϕ, or more briefly, the filter h. We define *the symbol* $\mathbf{h}$ of the filter h by

$$\mathbf{h}(\omega):=\frac{1}{\sqrt{2}}\sum_{n=-\infty}^{+\infty}h_n\exp(-\imath n\omega). \tag{3.5}$$

The reason we use Fourier series of $\{h_k\}_{k\in\mathbb{Z}}$ scaled by factor $1/\sqrt{2}$ instead of the original Fourier series to define the symbol is for the convenience of normalization of the formulas which will be discussed later. The filter *has a finite support* if there exist $k_1,k_2\in\mathbb{Z}$ such that $k_1<k_2$ and $h_k=0$ for all $k<k_1$ and $k>k_2$ $(k\in\mathbb{Z})$. If the filter has a finite support, then the summations in the scaling equation (3.4) and in the definition (3.5) of the symbol $\mathbf{h}$ are finite sums.

Theorem 3.6 ([25])**.** *A family of functions $\{\phi(\cdot-n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of the space V_0 if and only if there exist $A>0$ and $B>0$ such that*

$$\frac{1}{B}\le\sum_{k=-\infty}^{+\infty}|\hat{\phi}(\omega+2k\pi)|^2\le\frac{1}{A},\ \forall\ \omega\in[-\pi,\pi],$$

where $\hat{\phi}$ is the Fourier transform of ϕ, i.e. $\hat{\phi}(\omega)=\int_{-\infty}^{\infty}\phi(t)\exp(-\imath\omega t)dt$.

3.1.2 Scaling functions

Let f be a function in $L^2(\mathbb{R})$ and $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA. In every subspace V_j for every $j \in \mathbb{Z}$, f is optimally approximated by its orthogonal projection $P_{V_j}f$ into V_j. We need an orthonormal basis of V_j to compute the orthogonal projection.

We use $\delta_{m,n}$ to denote the Kronecker symbol, i.e.

$$\delta_{m,n} = \begin{cases} 1, & \text{if } m = n, \\ 0, & \text{if } m \neq n \end{cases}$$

Definition 3.7 (Orthogonal scaling function)**.** *We call a scaling function ϕ of an MRA $\{V_j\}_{j\in\mathbb{Z}}$ orthogonal if $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthogonal basis of V_j for all $j \in \mathbb{Z}$. Particularly, if these norms of the basis functions are one, we call ϕ to be orthonormal, i.e. for all j,*

$$\langle \phi_{j,m}, \phi_{j,n} \rangle = \delta_{m,n}.$$

In fact, from the properties of MRA, we know that if $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_j for any fixed $j_0 \in \mathbb{Z}$, then it is so for all j.

Lemma 3.8. *Let $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA and ϕ be a scaling function of it. Then ϕ is orthonormal if and only if its Fourier transform $\hat{\phi}$ satisfies*

$$\sum_{k=-\infty}^{+\infty} |\hat{\phi}(\omega + 2k\pi)|^2 = 1. \tag{3.6}$$

If a scaling function ϕ is not orthonormal, we can get an orthonormal scaling function φ by orthogonalization as following

$$\hat{\varphi}(\omega) = \frac{\hat{\phi}(\omega)}{\left(\sum_{k=-\infty}^{+\infty} |\hat{\phi}(\omega + 2k\pi)|^2\right)^{1/2}}.$$

We can easily check φ satisfies the orthonormal condition (3.6).

We also know that properties of scaling functions totally rely on the choice of the filter in the scaling equation (3.4). We will study the properties of scaling function ϕ with the filter h. By taking the Fourier transform of both sides of (3.4), we obtain

$$\hat{\phi}(\omega) = \mathbf{h}\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right)$$

We recursively use this relation to get

$$\hat{\phi}(\omega) = \left(\prod_{k=1}^{p} \mathbf{h}(2^{-k}\omega)\right) \hat{\phi}(2^{-p}\omega).$$

If $\hat{\phi}(\omega)$ is continuous (e.g. $\phi \in L^1$) at $\omega = 0$, then we have,

$$\hat{\phi}(\omega) = \left(\prod_{k=1}^{+\infty} \mathbf{h}(2^{-k}\omega)\right) \hat{\phi}(0).$$

Theorem 3.9 ([25]). *Let $\phi \in L^2(\mathbb{R}) \cap L^1(\mathbb{R})$ be an orthonormal scaling function, with the scaling equation*

$$\phi(\cdot) = \sqrt{2} \sum_{k=-\infty}^{+\infty} h_k \phi(2 \cdot -k), \tag{3.7}$$

Then the symbol $\mathbf{h}$ of the filter h satisfies

$$|\mathbf{h}(\omega)|^2 + |\mathbf{h}(\omega + \pi)|^2 = 1, \qquad \textit{for } \forall\ \omega \in \mathbb{R}, \tag{3.8}$$

and

$$\mathbf{h}(0) = 1. \tag{3.9}$$

Conversely, if $\mathbf{h}(\omega)$ is 2π periodic and continuously differentiable in a neighborhood of $\omega = 0$, if it satisfies (3.8) and (3.9) and if

$$\inf_{\omega \in [-\pi/2, \pi/2]} |\mathbf{h}(\omega)| > 0,$$

then

$$\hat{\phi}(\omega) = \prod_{k=1}^{+\infty} \mathbf{h}(2^{-k}\omega)$$

is the Fourier transform of a scaling function $\phi \in L^2(\mathbb{R})$.

Proof. See S. Mallat 1998 [25]. □

3.1.3 Orthogonal wavelets

Let f be an $L^2(\mathbb{R})$ function and $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA. Suppose W_j to be the orthogonal complement of V_j in V_{j+1}:

$$V_{j+1} = V_j \oplus W_j.$$

The orthogonal projection of f on V_{j+1} can be decomposed as the sum of the orthogonal projections on V_j and W_j:

$$P_{V_{j+1}} f = P_{V_j} f + P_{W_j} f.$$

Here the complement $P_{W_j} f$ is the detail information of f which can be described at the resolution level $j + 1$ but cannot be described at the level j. For $j < l$, we know that

$$V_{j+1} \subset V_l \text{ and } W_l \perp V_l,$$

thus,

$$W_l \perp V_{j+1} \text{ and } W_l \perp W_j \subset V_{j+1}.$$

Hence, for $j < l$,

$$V_l = \oplus_{k=j}^{l-1} W_k \oplus V_j. \tag{3.10}$$

Since $\{V_j\}_{j\in\mathbb{Z}}$ is an MRA, by letting $j \to -\infty$ and $l \to +\infty$ in (3.10), we get

$$L^2(\mathbb{R}) = \oplus_{j=-\infty}^{+\infty} W_j.$$

Therefore, f can be represented as the superposition of details of all resolution levels.

$$f = \sum_{j=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \langle f, \psi_{j,n} \rangle \psi_{j,n}.$$

Theorem 3.10. *Let ϕ be an orthonormal scaling function with the symbol $\mathbf{h}$. Let ψ be the function whose Fourier transform is*

$$\hat{\psi}(\omega) = \mathbf{g}\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right)$$

with

$$\mathbf{g}(\omega) = \exp(-\imath\omega)\mathbf{h}^*(\omega + \pi). \tag{3.11}$$

Let us denote

$$\psi_{j,n} = 2^{\frac{j}{2}} \psi(2^j \cdot -n), \qquad \text{for } j \in \mathbb{Z}.$$

Then, for any resolution level j, $\{\psi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthonormal basis of W_j, and for all $j \in \mathbb{Z}$, $\{\psi_{j,n}\}_{j,\,n\in\mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$.

Proof. See S. Mallat 1998 [25]. □

We call the function ψ *a wavelet*[3]. Sometimes people also call the wavelet ψ to be a *mother wavelet* and the scaling function ϕ to be a *father wavelet.* And we call W_j a *wavelet space* of resolution level j.

Definition 3.11 (Orthonormal wavelet)**.** *A wavelet ψ is orthonormal if $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$.*

Since $\psi \in W_0 \subset V_1$, we can represent ψ with the unique combination of the basis $\{\phi_{1,k}\}_{k\in\mathbb{Z}}$,

$$\psi(\cdot) = \sum_{n=-\infty}^{+\infty} g_n \phi(2 \cdot -n).$$

We call the sequence g_n *the filter of the wavelet of* ψ, or briefly *the filter* g, and the *symbol* $\mathbf{g}$ is the Fourier series of the filter g,

$$\mathbf{g}(\omega) = \frac{1}{\sqrt{2}} \sum_{n=-\infty}^{+\infty} g_n \exp(-\imath n\omega).$$

Apparently, properties of the wavelet ψ depend on its filter g and the scaling function ϕ. And since the filter g is constructed from the filter h of the scaling function ϕ, the main task in construction of wavelets is to construct the filter h.

[3] Here in this definition of wavelet, we are involved in the range of discontinuous wavelet transform. People define a zero average function in $L^2(\mathbb{R})$ to be a wavelet for the continuous wavelet transform. The requirement of the zero average is to guarantee the admissible condition which makes the reconstruction formula or the inverse wavelet transform possible, see Daubechies 1992 [8].

3.1.4 Constructing wavelets

When we want to construct a wavelet ψ, we mainly consider the number of vanishing moments of ψ and its support. We will investigate how the filter h is related on the requirement of the number of vanishing moments and support of ψ.

Vanishing moments:

Definition 3.12 (Vanishing moments)**.** *A function ψ has p vanishing moments if*

$$\int_{-\infty}^{+\infty} t^k \psi(t) dt = 0 \qquad \textit{for } 0 \leq k < p$$

In other words, the fact that ψ has p vanishing moments means that ψ is orthogonal to any polynomial of degree less than p.

Theorem 3.13 (Vanishing moments)**.** *Let ψ and ϕ be a wavelet and a scaling function, respectively, which are both orthonormal. And let $\mathbf{h}$ be the symbol of the filter h of ϕ. Suppose that*

$$|\psi(t)| = O((1+t^2)^{-p/2-1}) \textit{ and } |\phi(t)| = O((1+t^2)^{-p/2-1}).$$

The following four statements are equivalent:

(i) The wavelet ψ has p vanishing moments.

(ii) $\hat{\psi}(\omega)$ and its first $p-1$ derivatives are zeros at $\omega = 0$.

(iii) $\mathbf{h}(\omega)$ and its first $p-1$ derivatives are zeros at $\omega = \pi$.

(iv) for any $0 \leq k < p$,

$$q_k(\cdot) = \sum_{n=-\infty}^{+\infty} n^k \phi(\cdot - n) \quad \textit{is a polynomial of degree } k.$$

Proof. See S. Mallat 1998 [25]. □

Compact support:

In application, the number of non-zero coefficients of the filter of a scaling function directly affects the computational cost. The more the number of non-zero coefficients of the filter is, the greater is the cost of the computation. We know from the following lemma that the number of non-zero coefficients of the filter is related to the support size of the scaling function.

Lemma 3.14 (Compact support)**.** *The scaling function ϕ has a compact support if and only if the filter h has a compact support. Furthermore, their supports are equal.*

Proof. See I. Daubechies [7] and S. Mallat [25]. □

Note that, if a scaling function does not have a compact support, then the number of non-zero coefficients of the filter of the scaling function is infinite. We are interested in scaling functions or wavelets which have compact supports.

Daubechies compactly supported wavelets

Theorem 3.15. *Let* $h = \{h_0, h_1, \cdots, h_{2p-1}\}$ $(p \in \mathbb{N})$, *be a finite real sequence, whose symbol is* $\mathbf{h}$. *If* $\mathbf{h}$ *satisfies*

$$|\mathbf{h}(\omega)|^2 + |\mathbf{h}(\omega+\pi)|^2 = 1, \qquad \mathbf{h}(0) = 1,$$

and

$$\mathbf{h}(\omega) = \left(\frac{1+\exp(-\imath\omega)}{2}\right)^p t(\omega),$$

where t *is a trigonometric polynomial with* $|t(\omega)| \leq 2^{p-1/2}$ $(\forall\ \omega \in \mathbb{R})$. *Then there is a compactly supported orthonormal scaling function* $\phi \in L^2(\mathbb{R})$, *whose Fourier transform* $\hat{\phi}$ *is*

$$\hat{\phi}(\omega) = \frac{1}{\sqrt{2}} \prod_{k=1}^{+\infty} \mathbf{h}(2^{-k}\omega).$$

Proof. See Daubechies [7, 8] or Louis Maass and Rieder [23]. □

Daubechies (1988) has constructed a family of compactly supported orthonormal wavelets which has the minimum support size with a given number of vanishing moments [7].

Theorem 3.16 (Daubechies)**.** *Let* h *be the filter of a scaling function* ϕ *whose symbol* $\mathbf{h}(\omega)$ *has* p *zeros at* $\omega = \pi$, *then the filter* h *has at least* $2p$ *non-zero coefficients. And the filter of the scaling function of the Daubechies compactly supported wavelet of* p *vanishing moments has* $2p$ *non-zero coefficients.*

Proof. See I. Daubechies 1988 [7] or I. Daubechies 1992 [8]. □

We call this family of wavelets *Daubechies wavelets.* And we use DS_p and DW_p to denote the corresponding scaling function and wavelet.

Since Daubechies orthonormal compactly supported wavelets are asymmetric (see Figure 3.1) and the support sizes of wavelets are relatively large to obtain certain order of vanishing moments, compactly supported orthogonal wavelets are not optimal in applications. If we replace orthogonality with biorthogonality, we may obtain a more practical families of bases of $L^2(\mathbb{R})$.

3.2 Biorthogonal wavelets

The construction of compactly supported orthogonal wavelet with certain regularity is totally dependent on the design of the symbol $\mathbf{h}$ of the filter h. This is quite a *burden* to $\mathbf{h}$. By replacing orthogonality with biorthogonality, in which we introduce two more dual functions $\tilde{\phi}$, $\tilde{\psi}$, we may *relieve* the burden on a single $\mathbf{h}$ in the orthogonal case. These dual functions are called *dual scaling function* and *dual wavelet function* respectively, whose corresponding filters are $\tilde{h}$ and $\tilde{g}$. We have the following four refinement equations.

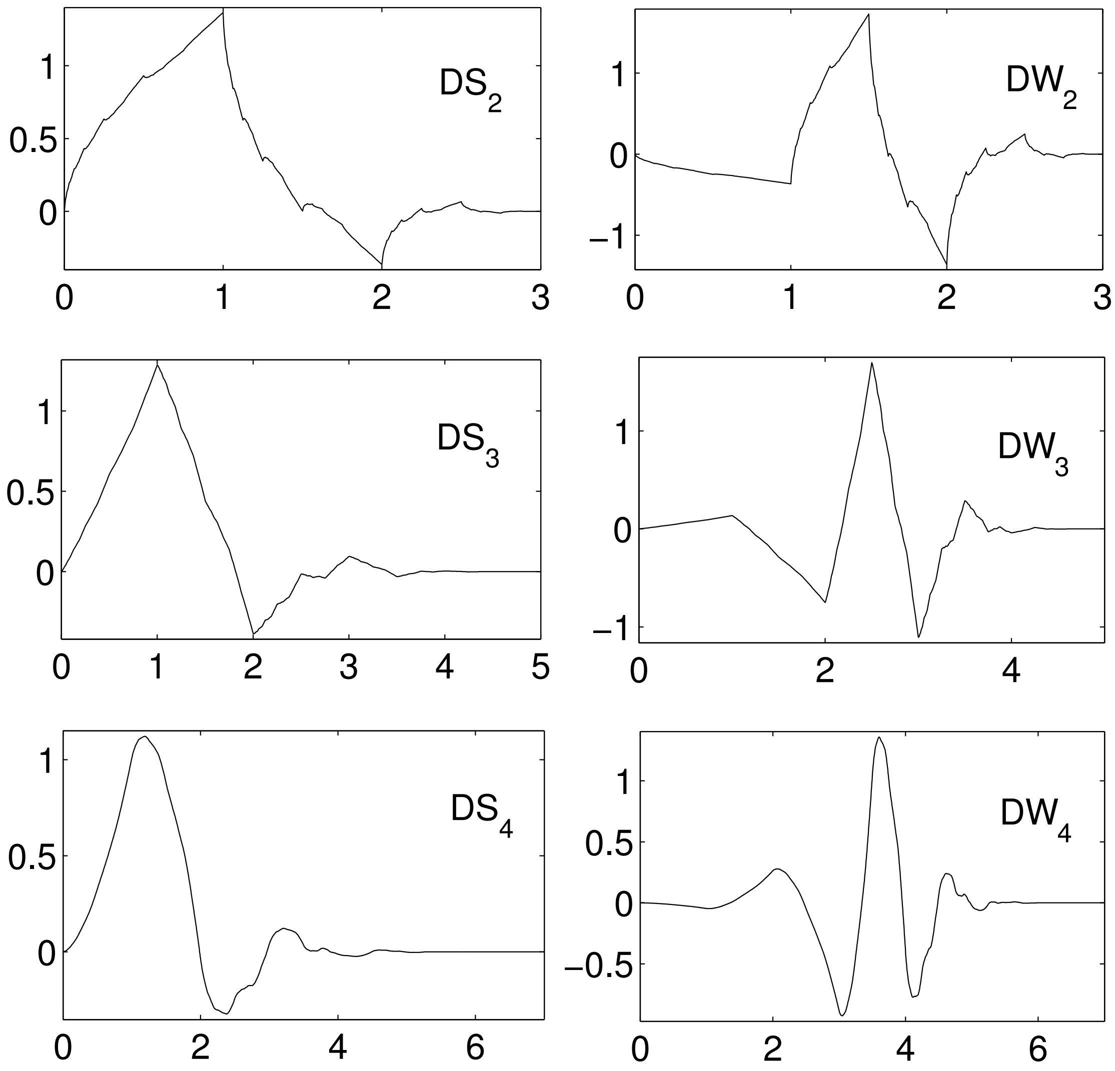


Figure 3.1: Daubechies scaling functions and wavelets of order 2, 3 and 4.

$$\phi(\cdot) = \sqrt{2} \sum_{n=-\infty}^{+\infty} h_n \phi(2 \cdot -n), \qquad \tilde{\phi}(\cdot) = \sqrt{2} \sum_{n=-\infty}^{+\infty} \tilde{h}_n \tilde{\phi}(2 \cdot -n)$$

$$\psi(\cdot) = \sqrt{2} \sum_{n=-\infty}^{+\infty} g_n \phi(2 \cdot -n), \qquad \tilde{\psi}(\cdot) = \sqrt{2} \sum_{n=-\infty}^{+\infty} \tilde{g}_n \tilde{\phi}(2 \cdot -n).$$

The biorthogonality requires us that

$$\begin{aligned} &\langle \phi(\cdot), \tilde{\phi}(\cdot - n) \rangle = \langle \psi(\cdot), \tilde{\psi}(\cdot - n) \rangle = \delta_{n,0} \text{ and} \\ &\langle \phi(\cdot), \tilde{\psi}(\cdot - n) \rangle = \langle \psi(\cdot), \tilde{\phi}(\cdot - n) \rangle = 0. \end{aligned} \tag{3.12}$$

Definition 3.17. *We call a set of scaling functions and wavelets, $\{\phi, \psi, \tilde{\phi}, \tilde{\psi}\}$, a family of biorthogonal scaling functions and wavelets if it satisfies (3.12).*

The filters $(h, g, \tilde{h}, \tilde{g})$ must satisfy

$$\sum_{k=-\infty}^{+\infty} h_k \tilde{h}_{k-2n} = \sum_{k=-\infty}^{+\infty} g_k \tilde{g}_{k-2n} = \delta_{n,0}, \tag{3.13a}$$

$$\sum_{k=-\infty}^{+\infty} h_k \tilde{g}_{k-2n} = \sum_{k=-\infty}^{+\infty} g_k \tilde{h}_{k-2n} = 0, \tag{3.13b}$$

for $\forall\ n \in \mathbb{Z}$.

We also have the biorthogonal condition in the form of symbol:

$$\overline{\mathbf{h}(\omega)}\tilde{\mathbf{h}}(\omega) + \overline{\mathbf{h}(\omega+\pi)}\tilde{\mathbf{h}}(\omega+\pi) = 1, \quad \overline{\mathbf{g}(\omega)}\tilde{\mathbf{g}}(\omega) + \overline{\mathbf{g}(\omega+\pi)}\tilde{\mathbf{g}}(\omega+\pi) = 1, \tag{3.14a}$$

$$\overline{\mathbf{h}(\omega)}\tilde{\mathbf{g}}(\omega) + \overline{\mathbf{h}(\omega+\pi)}\tilde{\mathbf{g}}(\omega+\pi) = 0, \quad \overline{\mathbf{g}(\omega)}\tilde{\mathbf{h}}(\omega) + \overline{\mathbf{g}(\omega+\pi)}\tilde{\mathbf{h}}(\omega+\pi) = 0, \tag{3.14b}$$

for $\forall\ \omega \in \mathbb{R}$.

Definition 3.18. *If a group of filters $(h, g, \tilde{h}, \tilde{g})$ satisfies (3.13), then we call it a family of biorthogonal filters. And if a group of symbols $(\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}})$ satisfies (3.14), then we call it a family of biorthogonal symbols.*

The following theorem of C. K. Chui (1992 [4]) tells us a general description on the dependence of the choice g and $\tilde{g}$ on h and $\tilde{h}$.

Theorem 3.19. *Let $\mathbf{h}$ and $\tilde{\mathbf{h}}$ be symbols of a scaling function and its dual and satisfy*

$$\overline{\mathbf{h}(\omega)}\tilde{\mathbf{h}}(\omega) + \overline{\mathbf{h}(\omega+\pi)}\tilde{\mathbf{h}}(\omega) = 1.$$

Then symbols $\mathbf{g}$ and $\tilde{\mathbf{g}}$ together with $\mathbf{h}$ and $\tilde{\mathbf{h}}$ compose a family of biorthogonal symbols if and only if there exists a function k, such that

$$k(\omega) = \sum_{n=-\infty}^{+\infty} c_n \exp(-\imath n\omega) \textit{ for } \omega \in \mathbb{R} \textit{ and } \sum_{n=-\infty}^{+\infty} |c_n| < \infty,$$

and satisfies that

$$\tilde{\mathbf{g}} = \exp(-\imath\omega)\overline{h(\omega+\pi)}k(2\omega) \textit{ and } \mathbf{g}(\omega) = \exp(-\imath\omega)\overline{\tilde{\mathbf{h}}(\omega+\pi)k^{-1}(2\omega)}.$$

We call a family of biorthogonal filters is *finite*, if each filter of the family is finite.

Let f be a function in $L^2(\mathbb{R})$. Let $\{V_j\}_{j\in\mathbb{Z}}$ be the MRA generated by ϕ and $\{W_j\}_{j\in\mathbb{Z}}$ be wavelets spaces generated by ψ. We have the corresponding dual MRA $\{\widetilde{V}_j\}_{j\in\mathbb{Z}}$ and dual wavelets spaces $\{\widetilde{W}_j\}_{j\in\mathbb{Z}}$ generated by $\tilde{\phi}$ and $\tilde{\psi}$, respectively. In this biorthogonal case, W_j's are not orthogonal to each other for different $j \in \mathbb{Z}$. Instead, we have

$$V_j \perp \widetilde{W}_j \text{ and } W_j \perp \widetilde{V}_j.$$

We can represent f with the basis $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$

$$f = \sum_{j=-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty} \langle f, \tilde{\psi}_{j,k}\rangle \psi_{j,k},$$

which is not an orthogonal decomposition.

The following work has been done by A. Cohen, I. Daubechies and J. -C. Feauveau (1992) [5], who first introduced biorthogonal wavelets and constructed compactly supported biorthogonal wavelets.

Theorem 3.20. *Let h and $\tilde{h}$ be filters with finite supports, whose symbol are $\mathbf{h}$ and $\tilde{\mathbf{h}}$, respectively. Suppose that*

$$\overline{\mathbf{h}(\omega)}\tilde{\mathbf{h}}(\omega) + \overline{\mathbf{h}(\omega+\pi)}\tilde{\mathbf{h}}(\omega+\pi) = 1, \qquad \mathbf{h}(0) = \tilde{\mathbf{h}}(0) = 1,$$

and

$$\mathbf{h}(\omega) = \left(\frac{1+\exp(-\imath\omega)}{2}\right)^{p} t(\omega) \quad \text{and} \quad \tilde{\mathbf{h}}(\omega) = \left(\frac{1+\exp(-\imath\omega)}{2}\right)^{\tilde{p}} \tilde{t}(\omega),$$

Here t and $\tilde{t}$ are trigonometric polynomials, and there exist k and $\tilde{k}$ in $\mathbb{N}$ such that

$$\max_{\omega\in\mathbb{R}} |\prod_{j=0}^{k-1} t(2^j\omega)| \le 2^{p-1/2} \quad \text{and} \quad \max_{\omega\in\mathbb{R}} |\prod_{j=0}^{\tilde{k}-1} \tilde{t}(2^j\omega)| \le 2^{\tilde{p}-1/2}.$$

Then, there exist biorthogonal scaling functions ϕ and $\tilde{\phi}$ whose corresponding filters are h and $\tilde{h}$, respectively. Moreover, by defining ψ and $\tilde{\psi}$ as

$$\psi = \sum_{n\in\mathbb{Z}} (-1)^n \tilde{h}_{1-n}\phi_{1,n} \quad \text{and} \quad \tilde{\psi} = \sum_{n\in\mathbb{Z}} (-1)^n h_{1-n}\tilde{\phi}_{1,n},$$

we have a family of biorthogonal wavelets functions $\{\phi, \psi, \tilde{\phi}, \tilde{\psi}\}$.

One can also obtain a new family of biorthogonal scaling functions and wavelets by a technique called *lifting* or *dual lifting.* This work has been done by W. Sweldens (1996) [32]. Moreover, I. Daubechies and W. Sweldens (1998) have shown that any family of finite biorthogonal filters can be obtained from any other family of finite biorthogonal filters by performing finite number of liftings and dual liftings on it [9].

So far, we have briefly introduced the basic theory on scaling functions and wavelets. It is convenient for us to choose a family of scaling functions which has interpolation property in solving differential equations.

3.3 Interpolating scaling functions

In this section, we will discuss the construction and the properties of the ISF's.

Construction of ISF's

In ISFM, we use uniform mesh and only involve scaling functions with the same resolution level. Therefore, here in this section we only introduce a method of construction of ISF's which constructs ISF's without involving the concept of wavelets.

Iterative interpolating process

We will construct ISF's on the real line with a method introduced by S. Dubuc (1986 [13]) and by G. Deslauries and S. Dubuc (1989 [10]). Let N be a positive integer. We interpolate the Kronecker sequence $\{\delta_{n,0}\}_{n\in\mathbb{Z}}$ at the integers to a function on the binary rationals by repeating the following process. For a given integer $j \geq 0$, if we have already obtained the values on all $k/2^j$, for all $k \in \mathbb{Z}$, then we interpolate the values at points $(k+1/2)/2^j$, for all $k \in \mathbb{Z}$, with the symmetric $2N$-points Lagrangian interpolations. Then we have values on all $k/2^{j+1}$. And we continue with the next resolution level and do the same symmetric $2N$-points Lagrangian interpolations. By repeating in this way, we will have values at any binary rationals, i.e., $k/2^j$, for $\forall\ k, j \in \mathbb{Z}$ and $j \geq 0$. We call this process *iterative interpolating process*. Thus we get a discrete function defined on a dense subset of $\mathbb{R}$. Since this function is uniformly continuous on the whole binary rationals, it can be uniquely extended to a function ϕ on $\mathbb{R}$. We also call this function *fundamental interpolating function of order N*, or briefly DD_N[4].

Properties of ISF(DD_N)

The construction with the iterative interpolation process tells us that $DD_N(N \in \mathbb{N})$ is even symmetric, i.e.

$$DD_N(-t) = DD_N(t), \quad t \in \mathbb{R}.$$

Theorem 3.21. *If p is a polynomial of degree less than $2N(N \in \mathbb{N})$, then p can be reproduced by the translates of DD_N.*

$$p(t) = \sum_{n=-\infty}^{+\infty} p(n) DD_N(t-n), \quad \textit{for } t \in \mathbb{R}.$$

Lemma 3.22. *Let $N \in \mathbb{N}$. Then DD_N vanishes outside $(-2N+1, 2N-1)$.*

Proof. We define a sequence $\{t_n\}_{n\geq 0}$ by the recurrence

$$t_{n+1} = t_n + (2N-1)/2^{n+1}, \quad n \in \mathbb{N},$$

with $t_0 = 0$. For a given $n \in \mathbb{N}$, we can observe from the process of iterative interpolation that DD_N vanishes outside $[-t_n, t_n]$ at the resolution level n, i.e., where the points are in the form of $k/2^n (k \in \mathbb{Z})$. We can easily compute that $t_n = 2N - 1 - (2N-1)/2^n$. By letting $n \to +\infty$, we know that DD_N vanishes outside $(-2N+1, 2N-1)$. □

Lemma 3.23. *$DD_N(N \in \mathbb{N})$ is continuously differentiable.*

[4] This term is named after Deslauriers and Dubuc.

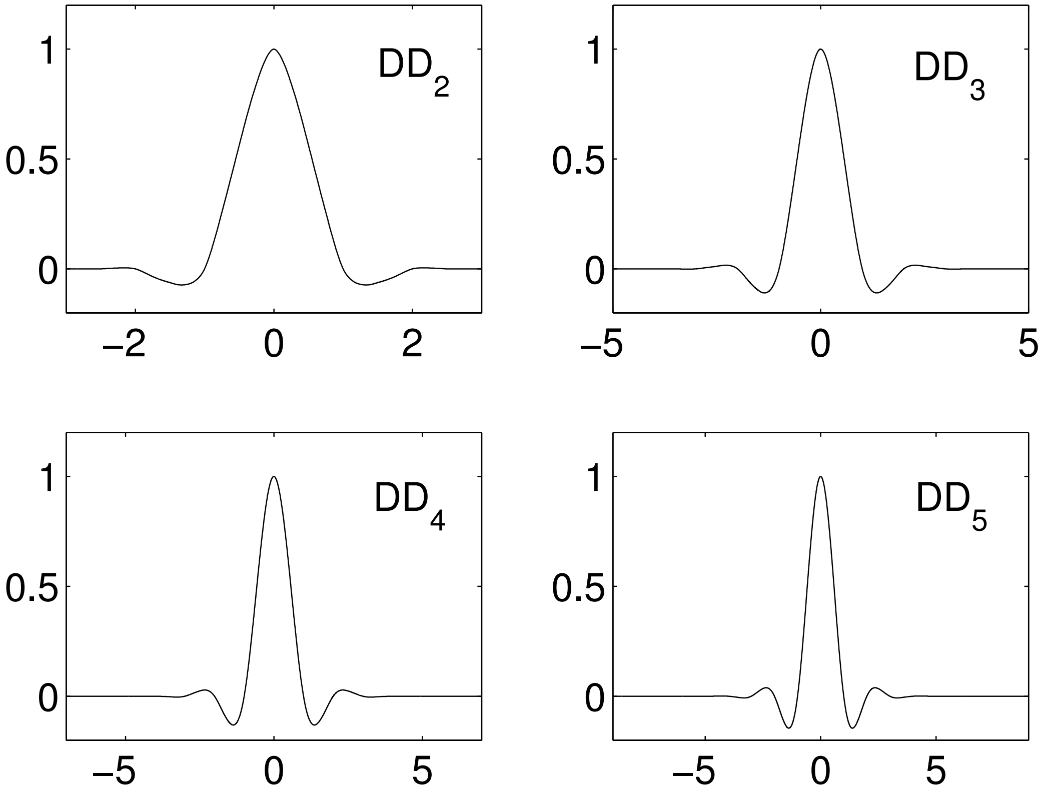


Figure 3.2: Deslauriers Dubuc functions of order from 2 to 5.

Proof. See S. Dubuc 1986 [13]. □

N. Saito and G. Beylkin have shown that $DD_N (N \in \mathbb{N})$ is the autocorrelation of Daubechies compactly supported orthogonal scaling function [31].

$$DD_N(t) = \int_{-\infty}^{+\infty} DS_N(x) DS_N(x-t)dx, \tag{3.15}$$

where DS_N is the Daubechies compactly supported orthogonal scaling function of order N, which yields the scaling equation,

$$DS_N(t) = \sum_{k=-2N+1}^{2N-1} h_k DS_N(2t-k), \quad t \in \mathbb{R}.$$

We also know that DD_N satisfies the scaling equation,

$$DD_N(t) = \sum_{k=-2N+1}^{2N-1} h_k^* DD_N(2t-k), \quad t \in \mathbb{R}.$$

One can also easily check the relation between the filters h^* and h from the autocorrelation equation (3.15),

$$h_n^* = \sum_{m=-\infty}^{+\infty} h_m h_{m-n}, \text{ for } \forall\ n \in \mathbb{Z}.$$

And since DD_N is interpolating, the filter h_k^* yields

$$h_k^* = DD_N(k/2).$$

Now we decompose any data set $\{f_i\}_{i\in\mathbb{Z}}$ as:

$$f_i = \sum_{j=-\infty}^{+\infty} f_j \delta_{i,j}, \tag{3.16}$$

where δ is the Kronecker symbol. Then the interpolation f of $\{f_i\}_{i\in\mathbb{Z}}$ on the whole $\mathbb{R}$ is combination of translates of DD_N.

$$f(t) = \sum_{j=-\infty}^{+\infty} f_j DD_N(t-j), \quad t \in \mathbb{R}.$$

Since DD_N is interpolating, we know that $f_j = f(j)$. We apply this relation on the data set $\{DD_N(j/2)\}_{j\in\mathbb{Z}}$ to have

$$DD_N(t/2) = \sum_{j=-\infty}^{+\infty} DD_N(j/2) DD_N(t-j), \quad t \in \mathbb{R}. \tag{3.17}$$

Since the DD_N has a compact support, the summation in (3.17) is finite. And since DD_N vanishes outside $(-2N+1, 2N+1)$, we know that $DD_N(j/2) = 0$ for $|j| \geq 4N-2$. Therefore

$$DD_N(t/2) = \sum_{j=-4N+3}^{4N-3} DD_N(j/2) DD_N(t-j), \text{ for } t \in \mathbb{R}. \tag{3.18}$$

We know from the interpolating property that $DD_N(j/2) = 0$, for even $j \in \mathbb{Z}$, hence, we only need to compute $DD_N(j/2)$ for odd integer j. And these can be computed using $2N$-points Lagrangian interpolations. One can easily check that

$$DD_N\left(\frac{2j+1}{2}\right) = (-1)^{N-j} \frac{\prod_{k=0}^{2N-1}(k-N+1/2)}{(j+1/2)(N-j-1)!(N+j)!}, \text{ for } -N \leq j < N. \tag{3.19}$$

If $N = 2$, we have

$$DD_2(\pm 1/2) = 9/16,\ DD_2(\pm 3/2) = -1/16 \text{ and } DD_2(\pm(2j+1)/2) = 0, \text{ for } j \geq 2.$$

Next, we will compute derivatives of DD_2 at half integer points, $j + 1/2$, which will be needed in ISFM. We know DD_2 has a finite support. And from the symmetry of DD_2, we have the anti-symmetry of the derivatives. Thus we only need to calculate $DD_2'(1/2)$, $DD_2'(3/2)$ and $DD_2'(5/2)$. We rewrite the equation (3.18), since we have computed its coefficients,

$$DD_2\left(\frac{t}{2}\right) = -\frac{1}{16}DD_2(t+3)+\frac{9}{16}DD_2(t+1)+DD_2(t)+\frac{9}{16}DD_2(t-1)-\frac{1}{16}DD_2(t-3), \quad t \in \mathbb{R}. \tag{3.20}$$

We differentiate both sides of (3.20) to have

$$\frac{1}{2}DD_2'\left(\frac{t}{2}\right) = -\frac{1}{16}DD_2'(t+3)+\frac{9}{16}DD_2'(t+1)+DD_2'(t)+\frac{9}{16}DD_2'(t-1)-\frac{1}{16}DD_2'(t-3), \quad t \in \mathbb{R}. \tag{3.21}$$

In order to compute $DD_2'(1/2)$, $DD_2'(3/2)$ and $DD_2'(5/2)$ from equation (3.21), we need first to compute $DD_2'(1)$ and $DD_2'(2)$.

We assume that f is the extension of a discrete data set $\{f_i\}_{i\in\mathbb{Z}}$ by iterative interpolation process with 4-points Lagrangian interpolations. Then we can write f as

$$f(t) = \sum_{j=-\infty}^{+\infty} f_j DD_2(t-j). \tag{3.22}$$

Differentiating (3.22) we obtain

$$f'(t) = \sum_{j=-\infty}^{+\infty} f_j DD_2'(t-j).$$

Taking $t = 0$, and since DD_2 vanishes outside the interval $(-3, 3)$ and $f_j = f(j)$, we have

$$f'(0) = -DD_2'(1)[f(1) - f(-1)] - DD_2'(2)[f(2) - f(-2)]. \tag{3.23}$$

Lemma 3.24.

$$DD_2'(1) = -2/3 \textit{ and } DD_2'(2) = 1/12.$$

Proof. This proof has been given by S. Dubuc 1986 [13].

We set $p_n = [f(2^{-n}) - f(-2^{-n})]/(2\cdot 2^{-n})$. The iterative interpolation process tells us that

$$f(-2^{-n}) = -1/16f(-4\cdot 2^{-n}) + 9/16f(-2\cdot 2^{-n}) + 9/16f(0) - 1/16f(2\cdot 2^{-n})$$
$$f(2^{-n}) = -1/16f(-2\cdot 2^{-n}) + 9/16f(0) + 9/16f(2\cdot 2^{-n}) - 1/16f(4\cdot 2^{-n})$$

Hence,

$$\begin{aligned} p_n =&[f(-4\cdot 2^{-n}) - 10f(-2\cdot 2^{-n}) + 10f(2\cdot 2^{-n}) - f(4\cdot 2^{-n})]/(32\cdot 2^{-n}) \\ =&5/4[f(2\cdot 2^{-n}) - f(-2\cdot 2^{-n})]/[2\cdot(2\cdot 2^{-n})] \\ &- 1/4[f(4\cdot 2^{-n}) - f(-4\cdot 2^{-n})]/[2\cdot(4\cdot 2^{-n})] \\ =&5/4p_{n-1} - 1/4p_{n-2}. \qquad \text{for } n \geq 1. \end{aligned} \tag{3.24}$$

The general solution of this difference equation (3.24) is

$$p_n = c_1 + c_2 4^{-n}.$$

We use the initial condition

$$p_{-1} = c_1 + 4c_2 = [f(2) - f(-2)]/4 \text{ and } p_0 = c_1 + c_2 = [f(1) - f(-1)]/2. \tag{3.25}$$

to compute c_1 and c_2, i.e.

$$c_1 = 2/3[f(1) - f(-1)] - 1/12[f(2) - f(-2)], \tag{3.26a}$$

$$c_2 = -1/16[f(1) - f(-1)] + 1/12[f(2) - f(-2)]. \tag{3.26b}$$

If we substitute (3.26) into (3.25) and let $n \to +\infty$, then, we have

$$f'(0) = 2/3[f(1) - f(-1)] - 1/12[f(2) - f(-2)]. \tag{3.27}$$

This equation (3.27) does not depend on the choice of f. Now compare this with (3.23), then we get $DD_2'(1) = -2/3$ and $DD_2'(2) = 1/12$. □

Hence, by substituting these into (3.21), we can obtain that $DD_2'(1/2) = -59/48$, $DD_2'(3/2) = -3/32$ and $DD_2'(5/2) = -1/96$.

Now if $N = 3$, by the formula (3.19), we know that

$$DD_3(\pm 1/2) = 75/128, \ DD_3(\pm 3/2) = -25/256 \text{ and } DD_3(\pm 5/2) = 3/256.$$

One can verify that the method used in the proof of the lemma above cannot be generalized for any N.

G. Deslauriers and S. Dubuc (1989) [10] have provided a beautiful technique on computations of these derivatives for $N = 3$, which may be generalized for the higher N.

Lemma 3.25.

$$DD_3'(1) = -272/365, \ DD_3'(2) = 53/365, \ DD_3'(3) = -16/1095 \textit{ and } DD_3'(4) = -1/2920.$$

Proof. We can reproduce any polynomial of degree less than 6 with translates of DD_3. Let us consider a polynomial $p(t) = t(t^2 - 1)(t^2 - 4)$. We know that p can be reproduced by DD_3,

$$p(t) = \sum_{n \in \mathbb{Z}} p(n) DD_3(t - n).$$

Since we know that p vanishes at $t = 0, \pm 1$ and ± 2, and $DD_3'(t) = 0$ for $|t| \geq 5$, we have

$$p'(0) = -2p(3)DD_3'(3) - 2p(4)DD_3'(4),$$

In the same way, if we consider other polynomials $q(t) = t(t^2 - 1)(t^2 - 9)$ and $r(t) = t(t^2 - 4)(t^2 - 9)$, we obtain

$$q'(0) = -2q(2)DD_3'(2) - 2q(4)DD_3'(4),$$

and

$$r'(0) = -2r(1)DD_3'(1) - 2r(4)DD_3'(4).$$

So far, we have three equations for four unknowns. If we consider the equation (3.18), and substitute t with $2s$, then we have

$$DD_3(s) = \sum_{j=-9}^{9} DD_3(j/2)DD_3(2s-j). \tag{3.28}$$

We differentiate both sides of (3.28) and take $s = 4$, then we have

$$DD_3'(4) = 2DD_3(5/2)DD_3'(3) = 3DD_3'(3)/128.$$

Therefore, we can get $DD_3'(1)$, $DD_3'(2)$, $DD_3'(3)$ and $DD_3'(4)$ by solving these four linear equations. □

With these derivatives on integer points ready, we can also compute $DD_3'(\pm j/2)$ by differentiating (3.18), for $j = 1, 3, 5, 7, 9$. See the table 3.1 and 3.2.

Table 3.1: Derivative filters $DD_N'(-i)$.

i	$N=2$	$N=3$	$N=4$
1	2/3	272/365	39296/49553
2	−1/12	−53/365	−76113/396424
3		16/1095	1664/49553
4		1/2920	−2645/1189272
5			−128/743295
6			1/1189272

Table 3.2: Derivative filters $DD_N'(-i-1/2)$.

i	$N=2$	$N=3$	$N=4$
0	59/48	120707/93440	266099391/202969088
1	−3/32	−76883/560640	−189991331/1217814528
2	1/96	1075/37376	63928787/1522268160
3		−1297/373760	−1505623/173973504
4		3/373760	1011845/1217814528
5			6637/608907264
6			−5/1217814528

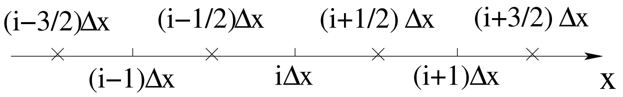


Figure 3.3: Yee's lattice for 1D.

3.4 Numerical approximations of the spatial derivatives with ISFM

Assume there is a uniform Yee's Lattice as in the figure 3.3. Where Δx is the smallest mesh size and $i\Delta x$'s ($i \in \mathbb{Z}$) are grid points and $(i+1/2)\Delta x$'s ($i \in \mathbb{Z}$) are midpoints of grid points.

Let f be a function which is discretized on midpoints of grid points, $(i+1/2)\Delta x$, for $i \in \mathbb{Z}$. We will approximate $f'(0)$ with the given values $\{f_{i+1/2} = f((i+1/2)\Delta x)\}_{i\in\mathbb{Z}}$ of f at midpoints, i.e., $(i+1/2)\Delta x$, $i \in \mathbb{Z}$. We approximate the function f with a function $\widetilde{f}_{\Delta x}$, which is a linear combination of translates of contracted ISF's.

$$\widetilde{f}_{\Delta x}(x) = \sum_{i\in\mathbb{Z}} f_{i+1/2} DD_N(x/\Delta x - i - 1/2), \quad \text{for } t \in \mathbb{R}, \tag{3.29}$$

where $f_{i+1/2} = f((i+1/2)\Delta x)$.

D. L. Donoho (1992 [12]) has shown that $\widetilde{f}_{\Delta x}$ converges to f in $C^\infty(\mathbb{R})$ as Δx goes to 0. Here we state two lemmas from [12].

Lemma 3.26. *Let $N \in \mathbb{N}$. And let V_j be a space spanned by $\{(DD_N)_{j,k}\}_{k\in\mathbb{Z}}$, for any non-negative integer j. Then the following statements are true:*

- *For any $f \in V_j$,*
$$f = \sum_k f(2^{-j}k)/2^{j/2}(DD_N)_{j,k}.$$
- *We have the inclusion*
$$V_j \subset V_{j+1}.$$
- *All the polynomials of degree less then $2N$ are in V_j.*

For a given function $f \in C_0(\mathbb{R})$, we define a projection $P_j f$ of f on $V_j (j \geq 0)$, i.e.,

$$V_j = \overline{span\{(DD_N)_{j,k} \mid k \in \mathbb{Z}\}},$$

as

$$P_j f := \sum_k f(2^{-j}k)/2^{j/2}(DD_N)_{j,k}.$$

Note that $P_j f$ is not an orthogonal projection since $(DD_N)_{j,k}$'s ($k \in \mathbb{Z}$) are not orthogonal.

Lemma 3.27. *If a function $f \in C_0(\mathbb{R})$, then*

$$\|f - P_j f\|_\infty \to 0, \ \text{as } j \to +\infty.$$

Differentiate both sides of (3.29), and put $x = 0$, then we have

$$\widetilde{f}'_{\Delta x}(0) = \frac{\sum_i f_{i+1/2} DD'_N(-i-1/2)}{\Delta x} \tag{3.30}$$

Since DD_N has a compact support, there exists an integer l_0 such that $DD'_N(-i-1/2) = 0$ for $|i+1/2| > l_0$ ($i \in \mathbb{Z}$). For the simplification, we use $a(i)$ to denote $DD'_N(-i-1/2)$, then we have

$$\widetilde{f}'_{\Delta x}(0) = \frac{\sum_{i=-l_0}^{l_0-1} f_{i+1/2} a(i)}{\Delta x}. \tag{3.31}$$

This is a general formula for ISFM with Yee's scheme. The higher the order N is, the larger is the compact support of DD_N, hence, the more is the number of terms in the sum of (3.31).

Now, in the TM$_y$ mode Maxwell's equations systems we approximate the time derivatives with central finite difference of second order, and the spatial derivatives with DD_N ($N \in \mathbb{N}$). Then, we have the following difference system with ISFM. The initial conditions and the boundary conditions are the same to that of Yee's FDTD method and we will not repeat them here.

$$\mathcal{B}_x|_{i,\,j+1/2}^{n+1/2} = \mathcal{B}_x|_{i,\,j+1/2}^{n-1/2} + \frac{\Delta t}{\Delta z} \sum_{l=-l_0}^{l_0-1} a(l) \mathcal{E}_y|_{i,\,j+l+1}^{n} \tag{3.32a}$$

$$\mathcal{H}_x|_{i,\,j+1/2}^{n+1/2} = \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{H}_x|_{i,\,j+1/2}^{n-1/2} + \frac{1}{\mu_0} \left(\frac{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{B}_x|_{i,\,j+1/2}^{n+1/2} - \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{B}_x|_{i,\,j+1/2}^{n-1/2} \right) \tag{3.32b}$$

$$\mathcal{B}_z|_{i+1/2,\,j}^{n+1/2} = \mathcal{B}_z|_{i+1/2,\,j}^{n-1/2} - \frac{\Delta t}{\Delta x} \sum_{l=-l_0}^{l_0-1} a(l) \mathcal{E}_y|_{i+l+1,\,j}^{n} \tag{3.32c}$$

$$\mathcal{H}_z|_{i+1/2,\,j}^{n+1/2} = \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \mathcal{H}_z|_{i+1/2,\,j}^{n-1/2} + \frac{1}{\mu_0} \left(\frac{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \mathcal{B}_z|_{i+1/2,\,j}^{n+1/2} - \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \mathcal{B}_z|_{i+1/2,\,j}^{n-1/2} \right) \tag{3.32d}$$

$$\widetilde{\mathcal{D}_y}|_{i,\,j}^{n+1} = \frac{1 - \dfrac{\sigma_x \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \widetilde{\mathcal{D}_y}|_{i,\,j}^{n} + \frac{\Delta t}{1 + \dfrac{\sigma_x \Delta t}{2\varepsilon_0}} \sum_{l=-l_0}^{l_0-1} a(l) \left(\frac{\mathcal{H}_x|_{i,\,j+l+1/2}^{n+1/2}}{\Delta z} - \frac{\mathcal{H}_z|_{i+l+1/2,\,j}^{n+1/2}}{\Delta x} \right) \tag{3.32e}$$

$$\mathcal{E}_y|_{i,\,j}^{n+1} = \frac{1 - \dfrac{\sigma_z \Delta t}{2\varepsilon_0}}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \mathcal{E}_y|_{i,\,j}^{n} + \frac{1}{1 + \dfrac{\sigma_z \Delta t}{2\varepsilon_0}} \frac{1}{\varepsilon} \left(\widetilde{\mathcal{D}_y}|_{i,\,j}^{n+1} - \widetilde{\mathcal{D}_y}|_{i,\,j}^{n} \right), \tag{3.32f}$$

where $a(i) = DD'_N(-i-1/2)$. And l_0 is an integer such that $[-l_0, l_0-1]$ is the support of the filter $\{a(l)\}_{l\in\mathbb{Z}}$.

3.5 Numerical dispersion and stability

We consider the difference system (3.32) in the main computational domain Ω (Figure 1.3) only, where all the lossy factors, σ's, are zero and no auxiliary variable such as $\mathcal{B}$ or $\widetilde{\mathcal{D}}$ is needed, then the system simplifies into:

$$\mathcal{H}_x|_{i,\,j+1/2}^{n+1/2} = \mathcal{H}_x|_{i,\,j+1/2}^{n-1/2} + \frac{\Delta t}{\mu\Delta z}\sum_{l=-l_0}^{l_0-1} a(l)\mathcal{E}_y|_{i,\,j+l+1}^{n}, \tag{3.33a}$$

$$\mathcal{H}_z|_{i+1/2,\,j}^{n+1/2} = \mathcal{H}_z|_{i+1/2,\,j}^{n-1/2} - \frac{\Delta t}{\mu\Delta x}\sum_{l=-l_0}^{l_0-1} a(l)\mathcal{E}_y|_{i+l+1,\,j}^{n}, \tag{3.33b}$$

$$\mathcal{E}_y|_{i,\,j}^{n+1} = \mathcal{E}_y|_{i,\,j}^{n} + \frac{\Delta t}{\varepsilon}\sum_{l=-l_0}^{l_0-1} a(l)\left(\frac{\mathcal{H}_x|_{i,\,j+l+1/2}^{n+1/2}}{\Delta z} - \frac{\mathcal{H}_z|_{i+l+1/2,\,j}^{n+1/2}}{\Delta x}\right). \tag{3.33c}$$

3.5.1 Numerical dispersion

Lemma 3.28. *If a pair of real numbers $\tilde{k}_x$ and $\tilde{k}_z$ satisfies*

$$\left[\frac{1}{v\Delta t}\sin\left(\frac{\omega\Delta t}{2}\right)\right]^2 = \left[\frac{1}{\Delta x}\sum_{l=-l_0}^{l_0-1} a(l)\sin\left(\frac{\tilde{k}_x(l+1/2)\Delta x}{2}\right)\right]^2 + \left[\frac{1}{\Delta z}\sum_{l=-l_0}^{l_0-1} a(l)\sin\left(\frac{\tilde{k}_z(l+1/2)\Delta z}{2}\right)\right]^2, \tag{3.34}$$

where $v = 1/\sqrt{\mu\varepsilon}$, and $a(l)$'s are derivative filters in Table 3.2, then the following set (3.35) is a solution to the difference system (3.33),

$$\mathcal{H}_x|_{I,\,J+1/2}^{n} = -\frac{\Delta t}{\mu\Delta z}\sum_{l=-l_0}^{l_0-1} a(l)\frac{\sin\left(\tilde{k}_z(l+1/2)\Delta z/2\right)}{\sin\left(\omega\Delta t/2\right)} \exp(\imath(\omega\, n\Delta t - \tilde{k}_x I\Delta x - \tilde{k}_z(J+1/2)\Delta z)), \tag{3.35a}$$

$$\mathcal{H}_z|_{I+1/2,\,J}^{n} = \frac{\Delta t}{\mu\Delta x}\sum_{l=-l_0}^{l_0-1} a(l)\frac{\sin\left(\tilde{k}_x(l+1/2)\Delta x/2\right)}{\sin\left(\omega\Delta t/2\right)} \exp(\imath(\omega\, n\Delta t - \tilde{k}_x(I+1/2)\Delta x - \tilde{k}_z J\Delta z)), \tag{3.35b}$$

$$\mathcal{E}_y|_{I,\,J}^{n} = \exp(\imath(\omega\, n\Delta t - \tilde{k}_x I\Delta x - \tilde{k}_z J\Delta z)). \tag{3.35c}$$

Conversely, if there exist a pair of real numbers $\tilde{k}_x$ and $\tilde{k}_z$ such that (3.35) is a solution to the difference system (3.33), then $\tilde{k}_x$ and $\tilde{k}_z$ yield (3.34).

Proof. The proof of this lemma is just a straightforward substitution of (3.35) into the system (3.33). We omit the detail here. □

The analysis of numerical dispersion with higher order method such as ISFM is not as simple as that of FDTD, since it is difficult to analytically solve for numerical wavenumbers, $\tilde{k}_x$ and $\tilde{k}_z$. E. M. Tentzeris, R. L. Robertson, J. F. Harvey and L. P. B. Katehi have used BisectionNewtonRaphson hybrid technique to solve this non-linear dispersion equation (3.34) with different type of wavelets, i.e., different $a(l)$, and provided comparison results with FDTD, see [36]. They showed that higher order schemes have less dispersion error than FDTD. We don't want to discuss it in detail here. According to our experience in numerical examples, the computation errors caused by numerical dispersion are less than that of FDTD of second order.

3.5.2 Numerical stability

The stability analysis for the ISFM is similar to that of Yee's FDTD scheme, see subsection 2.3.2. We keep all the notations of the subsection 2.3.2 except $\widetilde{\nabla}$, which is the discretized gradient with respect to central finite difference of second order, and we use $\overline{\overline{\nabla}}$ instead of $\widetilde{\nabla}$, which is the discretized gradient with respect to ISFM difference scheme. Here we will have the following equation instead of (2.23),

$$\frac{\mathcal{V}|_{p,q,r}^{n+1/2} - \mathcal{V}|_{p,q,r}^{n-1/2}}{\Delta t} = \frac{j}{\sqrt{\mu\varepsilon}} \overline{\overline{\nabla}} \times \mathcal{V}|_{p,q,r}^{n}, \tag{3.36}$$

We substitute (2.24) into (3.36) and obtain

$$\frac{\alpha^{1/2} - \alpha^{-1/2}}{\Delta t} V_0 = -\frac{2}{\sqrt{\mu\varepsilon}} (u_1, u_2, u_3) \times V_0,$$

where

$$u_1 = \left[\frac{1}{\Delta x} \sum_{l=0}^{l_0 - 1} a(l) \sin(k_x(l + 1/2)\Delta x/2) \right]^2,$$

$$u_2 = \left[\frac{1}{\Delta y} \sum_{l=0}^{l_0 - 1} a(l) \sin(k_y(l + 1/2)\Delta y/2) \right]^2,$$

$$u_3 = \left[\frac{1}{\Delta z} \sum_{l=0}^{l_0 - 1} a(l) \sin(k_z(l + 1/2)\Delta z/2) \right]^2$$

We also get a similar form to (2.26),

$$\frac{\alpha^{1/2} - \alpha^{-1/2}}{\Delta t} V_0 = \frac{2}{\sqrt{\mu\varepsilon}} B V_0,$$

where

$$B = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}.$$

Then, as in Subsection 2.3.1, by solving eigenvalues of the matrix B, we obtain,

$$\left(\frac{\sqrt{\mu\varepsilon}(\alpha^{1/2} - \alpha^{-1/2})}{2\Delta t} \right)^2 + u_1^2 + u_2^2 + u_3^2 = 0.$$

The derivation here is not different from that in the former chapter. We get

$$v\Delta t \sqrt{u_1^2 + u_2^2 + u_3^2} \leq 1. \tag{3.37}$$

We require that the inequality (3.37) should hold for all possible k_x, k_y and k_z, thus we have the stability condition of the Yee's ISFM scheme:

$$\Delta t \leq \frac{1}{v \sum\limits_{l=0}^{l_0-1} |a(l)| \sqrt{\dfrac{1}{(\Delta x)^2} + \dfrac{1}{(\Delta y)^2} + \dfrac{1}{(\Delta z)^2}}}.$$

For 2D case we have

$$\Delta t \leq \left[v \left(\sum_{l=0}^{l_0-1} |a(l)| \right) \sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta z)^2}} \right]^{-1} \tag{3.38}$$

this is generalization of the stability analysis of FDTD Yee's scheme, where differentiation filters $a(-1) = -1/2$ and $a(1) = 1/2$. Note that the stability conditions derived here are not only for the Yee's scheme, which is staggered and uncollocated, but also valid for the unstaggered and collocated scheme.

Chapter 4

Adaptive wavelet collocation method

In the ISFM, which was discussed in the previous chapter, we use a uniform grid to discretize field values. Only scaling functions of one resolution level are involved in the ISFM. In this chapter, we continue to discuss the adaptive wavelet collocation method (AWCM) (O. Vasilyev [40, 39]), which considers not only scaling functions of one resolution level but also wavelets of different resolution levels.

Let $\{\phi, \psi, \tilde{\phi}, \tilde{\psi}\}$ be a family of biorthogonal scaling functions and wavelets. And let $\{V_j\}_{j\in\mathbb{Z}}$ and $\{W_j\}_{j\in\mathbb{Z}}$ be the corresponding MRA and the wavelet spaces. In general, W_j is not the orthogonal complement of V_j $(j \in \mathbb{Z})$ in V_{j+1}. Suppose that $j_0, j_1 \in \mathbb{N}$ $(j_0 < j_1)$. We know the following relation:

$$V_{j_1} = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \cdots \oplus W_{j_1-1}. \tag{4.1}$$

Note that " $\oplus$ " in (4.1) is a direct sum which is not necessarily orthogonal.

Let f be a function in $L^2(\mathbb{R})$. We use P_j and Q_j to denote projections of f into the subspaces V_j and W_j $(j \in \mathbb{Z})$ of $L^2(\mathbb{R})$, respectively. Then from (4.1) we have

$$P_{j_1} f = P_{j_0} f + \sum_{j=j_0}^{j_1-1} Q_j f. \tag{4.2}$$

We represent the R.H.S. of (4.2) with the basis functions $\{\phi_{j_0,k}\}_{k\in\mathbb{Z}}$ and $\{\psi_{j,m}\}_{j,m\in\mathbb{Z}}$, then we get

$$P_{j_1} f = \sum_{k\in\mathbb{Z}} \alpha_{j_0,k}\phi_{j_0,k} + \sum_{j=j_0}^{j_1-1} \sum_{m\in\mathbb{Z}} \beta_{j,m}\psi_{j,m},$$

where

$$\alpha_{j_0,k} = \langle f, \tilde{\phi}_{j_0,k}\rangle \text{ and } \beta_{j,m} = \langle f, \tilde{\psi}_{j,m}\rangle.$$

We call $\alpha_{j_0,k}$ *scaling coefficient* and $\beta_{j,m}$ *wavelet coefficient*. We may think that scaling coefficients $\alpha_{j_0,k}$ $(k \in \mathbb{Z})$ show *rough information* of $P_{j_1} f$ at the coarsest resolution level j_0 and the wavelet coefficients $\beta_{j,m}$ $(j = j_0, \cdots, j_1 - 1, m \in \mathbb{Z})$ show *detailed information* of $P_{j_1} f$ at various resolution levels from j_0 to $j_1 - 1$.

Our interest is in the wavelet coefficients. In the applications such as numerical solutions to partial differential equations (PDE's) or image compressions, we may approximate a function or signal by discarding some terms in its wavelet decomposition whose wavelet coefficients are *negligible*, i.e. the absolute values of the wavelet coefficients are less than a given thresholding tolerance value. We can obtain wavelet coefficients with fast wavelet transforms. In the AWCM, we will use the *lifted interpolating wavelet transform.* We can obtain the lifted interpolating wavelets by lifting a type of wavelets called *interpolating wavelets* or *Donoho wavelets.*

4.1 Interpolating wavelets

Let us first study about the interpolating wavelets introduced by Donoho 1992 [12].

For $N \in \mathbb{N}$, we define a function Do_N (named after Donoho) by

$$Do_N(\cdot) := 2DD_N(2\cdot -1).$$

Assume $j \in \mathbb{Z}$ $(j \geq 0)$. Define $W_j := \overline{span\{(Do_N)_{j,m} \,|\, m \in \mathbb{Z}\}}$. Note that,

$$(Do_N)_{j,m}(\cdot) = 2^{j/2}(Do_N)(2^j \cdot -m) = \sqrt{2}(DD_N)_{j+1,2m+1}(\cdot).$$

Hence, we know that

$$W_j \subset V_{j+1} = \overline{span\{(DD_N)_{j+1,k} \,|\, k \in \mathbb{Z}\}}.$$

Suppose we have two sequences $\{\alpha_{j,k}\}_{k\in\mathbb{Z}}$ and $\{\beta_{j,m}\}_{m\in\mathbb{Z}}$. Let f be a function constructed by

$$f = \sum_{k\in\mathbb{Z}} \alpha_{j,k}(DD_N)_{j,k} + \sum_{m\in\mathbb{Z}} \beta_{j,m}(Do_N)_{j,m}. \tag{4.3}$$

We know that $f \in V_{j+1}$, thus we can also represent f as

$$f = \sum_{k\in\mathbb{Z}} \alpha_{j+1,k}(DD_N)_{j+1,k}. \tag{4.4}$$

For a given integer j, we define a set of dyadic rationals of degree j,

$$\mathcal{K}_j := \{k/2^j \mid k \in \mathbb{Z}\}.$$

Then we have the nested relation,

$$\mathcal{K}_j \subset \mathcal{K}_{j+1}.$$

We define $\mathcal{M}_j := \mathcal{K}_{j+1} \setminus \mathcal{K}_j$, for $j \in \mathbb{Z}$. The grid set $\mathcal{K}_{j+1}$ of level $j+1$ is composed of two independent parts, $\mathcal{K}_j$ and $\mathcal{M}_j$. We call $\mathcal{K}_j$ a *coarse part* and $\mathcal{M}_j$ a *fine part* in the level $j+1$.

Now we consider the two forms (4.3) and (4.4) of representations of $f \in V_{j+1}$. From the definition of Do_N, we know that

$$(Do_N)_{j,m}(t) = 0, \qquad \text{for } t \in \mathcal{K}_j.$$

Therefore, we have

$$\sum_{k\in\mathbb{Z}} \alpha_{j+1,k}(DD_N)_{j+1,k}(t) = \sum_{k\in\mathbb{Z}} \alpha_{j,k}(DD_N)_{j,k}(t), \qquad \text{for } t \in \mathcal{K}_j.$$

Hence, we have

$$\alpha_{j,k} = \sqrt{2}\alpha_{j+1,2k}, \qquad \text{for } k \in \mathbb{Z}.$$

Let $t_0 \in \mathcal{M}_j$, i.e., $\exists\, k \in \mathbb{Z}$ s.t. $t_0 = (k+1/2)/2^j$. Substituting t_0 into (4.3) and (4.4), and using the interpolating property of DD_N, we have

$$\sqrt{2}\alpha_{j+1,2k+1} = 2^{-j/2}\sum_{m\in\mathbb{Z}} \alpha_{j,m}(DD_N)_{j,m}((k+1/2)/2^j) + \beta_{j,k}.$$

Hence, we have the following lemma.

Lemma 4.1. *Every $f \in V_{j+1}$ can be represented as (4.3), with*

$$\alpha_{j,k} = 2^{-\frac{j}{2}} f\left(\frac{k}{2^j}\right), \beta_{j,k} = 2^{-\frac{j}{2}}\left(f\left(\frac{k+\frac{1}{2}}{2^j}\right) - (P_j f)\left(\frac{k+\frac{1}{2}}{2^j}\right)\right).$$

This lemma shows that the wavelet coefficients $\beta_{j,k}$'s ($k \in \mathbb{Z}$) represent the difference of f and its approximation $P_j f$ in V_j.

Now we discuss the symbols of the family of biorthogonal scaling functions and wavelets. Let h and g be the filters of DD_N and Do_N respectively and $\mathbf{h}$ and $\mathbf{g}$ be their corresponding symbols. Since DD_N is interpolating, it is easy to check from its scaling equation that $h_{2k} = \delta_{k,0}/\sqrt{2}$, for all $k \in \mathbb{Z}$. In the symbol form we have

$$\mathbf{h}(\omega) + \mathbf{h}(\omega+\pi) = 1, \quad \omega \in \mathbb{R}. \tag{4.5}$$

Since $Do_N(t) = 2DD_N(2t-1)$, for $t \in \mathbb{R}$, we know that $g = \{g_0 = 0, g_1 = \sqrt{2}\}$. Therefore, $\mathbf{g}(\omega) = \exp(-\imath\omega)$, $\omega \in \mathbb{R}$. The biorthogonal condition (3.12) in matrix form is

$$\tilde{M}(\omega)M(\omega)^* = I, \tag{4.6}$$

where $M, \tilde{M} \in \mathbb{C}^{2\times 2}$, and

$$M(\omega) = \begin{bmatrix} \mathbf{h}(\omega) & \mathbf{h}(\omega+\pi) \\ \mathbf{g}(\omega) & \mathbf{g}(\omega+\pi) \end{bmatrix} \text{ and } \tilde{M}(\omega) = \begin{bmatrix} \tilde{\mathbf{h}}(\omega) & \tilde{\mathbf{h}}(\omega+\pi) \\ \tilde{\mathbf{g}}(\omega) & \tilde{\mathbf{g}}(\omega+\pi) \end{bmatrix},$$

and I is the 2×2 identity matrix. The notation "$*$" means conjugate transpose of a complex matrix.

Thus, we have

$$\tilde{M}(\omega) = (M(\omega)^*)^{-1} = \begin{bmatrix} 1 & 1 \\ \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)} & \exp(-\imath(\omega+\pi))\overline{\mathbf{h}(\omega+2\pi)} \end{bmatrix}.$$

Therefore, we can choose $\tilde{\mathbf{h}}(\omega) = 1$ and $\tilde{\mathbf{g}}(\omega) = \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)}$. Thus, the dual scaling function $\tilde{\phi}$ is the Dirac impulse at the origin and the dual scaling function $\tilde{\psi}$ is a linear combination of the shifted Dirac impulses. This biorthogonal family $\{DD_N, Do_N, \tilde{\phi}, \tilde{\psi}\}$ with its family of symbols $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$ is called *Donoho wavelets family.*

The Donoho wavelets family has some disadvantages.

1. Do_N does not have vanishing moments. Thus, it does not satisfy the admissible condition, which means it cannot generate a Riesz basis of $L^2(\mathbb{R})$.

2. The duals are not $L^2(\mathbb{R})$ functions.

W. Sweldens has introduced a smart technique, called *lifting scheme* 1996 [32], which can be used to construct better families of biorthogonal scaling functions and wavelets overcoming these disadvantages of Donoho wavelets family. This will be the main contents of the next section.

4.2 The lifting scheme

In this section we will introduce the lifting scheme developed by W. Sweldens (1996 [32] and 1998 [33]). We start with the following lemma.

Lemma 4.2. *If we have a family of biorthogonal symbols,* $\{\mathbf{h}, \mathbf{g}^0, \tilde{\mathbf{h}}^0, \tilde{\mathbf{g}}\}$*. Then, the new family of symbols,* $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$*, defined as the following,*

$$\mathbf{g}(\omega) = \mathbf{g}^0(\omega) - \mathbf{h}(\omega)s(2\omega), \tag{4.7a}$$

$$\tilde{\mathbf{h}}(\omega) = \tilde{\mathbf{h}}^0(\omega) + \tilde{\mathbf{g}}(\omega)\overline{s(2\omega)}, \tag{4.7b}$$

for $\omega \in \mathbb{R}$*, where* s *is a trigonometrical polynomial, is also a family of biorthogonal symbols.*

Proof. Our task is to check the biorthogonal condition in matrix form (4.6). Let M^0, $\tilde{M}^0$, M and $\tilde{M}$ be the corresponding 2×2 complex matrix functions, i.e.,

$$M^0(\omega) = \begin{bmatrix} \mathbf{h}(\omega) & \mathbf{h}(\omega+\pi) \\ \mathbf{g}^0(\omega) & \mathbf{g}^0(\omega+\pi) \end{bmatrix}, \ \tilde{M}^0(\omega) = \begin{bmatrix} \tilde{\mathbf{h}}^0(\omega) & \tilde{\mathbf{h}}^0(\omega+\pi) \\ \tilde{\mathbf{g}}(\omega) & \tilde{\mathbf{g}}(\omega+\pi) \end{bmatrix},$$

$$M(\omega) = \begin{bmatrix} \mathbf{h}(\omega) & \mathbf{h}(\omega+\pi) \\ \mathbf{g}(\omega) & \mathbf{g}(\omega+\pi) \end{bmatrix}, \ \tilde{M}(\omega) = \begin{bmatrix} \tilde{\mathbf{h}}(\omega) & \tilde{\mathbf{h}}(\omega+\pi) \\ \tilde{\mathbf{g}}(\omega) & \tilde{\mathbf{g}}(\omega+\pi) \end{bmatrix},$$

for $\omega \in \mathbb{R}$. From (4.7), we know that

$$\tilde{M} = S_1\tilde{M}^0, \text{ and } M = S_2M^0,$$

where S_1 and S_2 are 2×2 matrix funtions and

$$S_1(\omega) = \begin{bmatrix} 1 & \overline{s(2\omega)} \\ 0 & 1 \end{bmatrix} \text{ and } S_2(\omega) = \begin{bmatrix} 1 & 0 \\ -s(2\omega) & 1 \end{bmatrix}, \quad \omega \in \mathbb{R}.$$

We know that the old family of symbols is biorthogonal, i.e., $\tilde{M}^0(M^0)^* = I$, and it is easy to check that $S_1S_2^* = I$, therefore,

$$\tilde{M}(M)^* = S_1\tilde{M}^0(M^0)^*S_2^* = S_1(\tilde{M}^0(M^0)^*)S_2^* = S_1IS_2 = I.$$

Hence, the new family of symbols is also biorthogonal. □

Next we observe how these changes of symbols affect the corresponding functions. We see that the scaling function ϕ^0 does not change, i.e, $\phi = \phi^0$, because its symbol $\mathbf{h}^0$ does not change. However, the other three functions change. We consider the Fourier transforms of the refinement equations of these three functions.

$$\begin{aligned}\hat{\psi}(\omega) &= \mathbf{g}(\omega/2)\hat{\phi}^0(\omega/2)\\ &= \mathbf{g}^0(\omega/2)\hat{\phi}^0(\omega/2) - s(\omega)\mathbf{h}^0(\omega/2)\hat{\phi}^0(\omega/2)\\ &= \mathbf{g}^0(\omega/2)\hat{\phi}^0(\omega/2) - s(\omega)\hat{\phi}^0(\omega), \quad \omega \in \mathbb{R}.\end{aligned}$$

We assume that

$$s(\omega) = \sum_k s_k \exp(-\imath\omega), \quad s_k \in \mathbb{R},$$

then by the inverse Fourier transform, we have

$$\psi(t) = \sqrt{2}\sum_k g_k^0\phi^0(2t-k) - \sum_k s_k\phi^0(t-k), \quad t \in \mathbb{R}.$$

Similarly we have

$$\begin{aligned}\hat{\tilde{\phi}}(\omega) &= \tilde{\mathbf{h}}(\omega/2)\hat{\tilde{\phi}}(\omega/2)\\ &= \tilde{\mathbf{h}}^0(\omega/2)\hat{\tilde{\phi}}(\omega/2) + \overline{s(\omega)}\tilde{\mathbf{g}}^0(\omega/2)\hat{\tilde{\phi}}(\omega/2)\\ &= \tilde{\mathbf{h}}^0(\omega/2)\hat{\tilde{\phi}}(\omega/2) + \overline{s(\omega)}\hat{\tilde{\psi}}(\omega), \quad \omega \in \mathbb{R}.\end{aligned}$$

Thus, by the inverse Fourier transform, we have

$$\tilde{\phi}(t) = \sqrt{2}\sum_k \tilde{\mathbf{h}}_k^0\tilde{\phi}(2t-k) + \sum_k s_{-k}\tilde{\psi}(t-k), \quad t \in \mathbb{R}.$$

We also have

$$\hat{\tilde{\psi}}(\omega) = \tilde{\mathbf{g}}(\omega/2)\hat{\tilde{\phi}}(\omega/2), \quad \omega \in \mathbb{R}.$$

Therefore, we get

$$\tilde{\psi}(t) = \sqrt{2}\sum_k \tilde{g}_k\tilde{\phi}(2t-k), \quad t \in \mathbb{R}.$$

We have the following theorem.

Theorem 4.3. *Suppose that we have a family of biorthogonal scaling functions and wavelets* $\{\phi, \psi^0, \tilde{\phi}^0, \tilde{\psi}^0\}$. *Then a new family* $\{\phi, \psi, \tilde{\phi}, \tilde{\psi}\}$, *defined as,*

$$\begin{aligned}\psi(\cdot) &= \psi^0(\cdot) - \sum_k s_k\phi(\cdot - k),\\ \tilde{\phi}(\cdot) &= \sqrt{2}\sum_k \tilde{\mathbf{h}}_k^0\tilde{\phi}(2\cdot - k) + \sum_k s_{-k}\tilde{\psi}(\cdot - k),\\ \tilde{\psi}(\cdot) &= \sqrt{2}\sum_k \tilde{g}_k\tilde{\phi}(2\cdot - k),\end{aligned}$$

where $s_k \in \mathbb{R}$, *is biorthogonal.*

Remark 4.4. 1. This process of obtaining a new family of biorthogonal scaling functions and wavelets is called *lifting*.

2. Assume we have an initial family $\{\mathbf{h}^0, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}^0\}$ of biorthogonal symbols, then we can also get a new family $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$ of biorthogonal symbols defined by

$$\begin{aligned}\mathbf{h}(\omega) &= \mathbf{h}^0(\omega) + \mathbf{g}(\omega)\tilde{s}(2\omega),\\ \tilde{\mathbf{g}}(\omega) &= \tilde{\mathbf{g}}^0(\omega) - \tilde{\mathbf{h}}(\omega)\overline{\tilde{s}(2\omega)},\end{aligned}$$

where $\tilde{s}$ is a trigonometric polynomial. We call this process *dual lifting*.

3. The choice of trigonometric polynomials s and $\tilde{s}$ both in lifting and dual lifting processes plays an important role in improving the properties of the scaling functions and wavelets.

Example (Lazy wavelet)

We have a family of biorthogonal symbols simply defined as:

$$2\mathbf{h}^0(\omega) = \tilde{\mathbf{h}}(\omega) = 1 \quad \text{and} \quad \mathbf{g}(\omega) = 2\tilde{\mathbf{g}}^0(\omega) = \exp(-\imath\omega), \quad \omega \in \mathbb{R}.$$

People call this family $\{\mathbf{h}^0, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}^0\}$ of symbols *family of Lazy wavelet symbols*, see for example W. Sweldens 1996 [32] and S. Mallat 1998 [25].

We assume that a family $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$ is a Donoho wavelets family, i.e., $\mathbf{h}$ is a symbol of DD_N ($N \in \mathbb{N}$), and

$$\mathbf{g}(\omega) = \exp(-\imath\omega), \;\; \tilde{\mathbf{h}}(\omega) = 1, \;\text{and}\; \tilde{\mathbf{g}}(\omega) = \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)}, \quad \omega \in \mathbb{R}.$$

We will investigate how these two families $\{\mathbf{h}^0, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}^0\}$ and $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$ relate to each other.

Since $\mathbf{h}$ is a symbol of an ISF, we know that $h_{2k} = \delta_{k,0}/\sqrt{2}$ and h is finite. And also we know that $h_{2k+1} = DD_N((2k+1)/2)/\sqrt{2}$. Hence, there exists a trigonometric polynomial $\tilde{s}$ such that

$$\tilde{s}(2\omega) = \exp(\imath\omega)\left(\mathbf{h}(\omega) - 1/2\right), \quad \omega \in \mathbb{R}.$$

Then we have

$$\begin{aligned}\mathbf{h}(\omega) &= 1/2 + \exp(-\imath\omega)\tilde{s}(2\omega)\\ &= \mathbf{h}^0(\omega) + \mathbf{g}(\omega)\tilde{s}(2\omega), \quad \omega \in \mathbb{R}.\end{aligned} \tag{4.8}$$

And by using the fact that $\mathbf{h}(\omega) + \mathbf{h}(\omega+\pi) = 1$ ($\omega \in \mathbb{R}$), (see (4.5)), we have

$$\begin{aligned}\tilde{\mathbf{g}}(\omega) &= \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)}\\ &= \exp(-\imath\omega)\overline{1 - \mathbf{h}(\omega)}\\ &= 1/2\exp(-\imath\omega) + \exp(-\imath\omega)\overline{1/2 - \mathbf{h}(\omega)}\\ &= \tilde{\mathbf{g}}^0(\omega) - \tilde{\mathbf{h}}(\omega)\overline{\tilde{s}(2\omega)}, \quad \omega \in \mathbb{R}.\end{aligned}$$

Therefore, we can see that the family of Donoho wavelets can be obtained by performing a dual lifting scheme on the family of Lazy wavelet.

Lifting Donoho wavelets family

Now we will continue to lift the family of Donoho wavelets in order to improve the properties of wavelet functions. We start with the family of the symbols of Donoho wavelets.

We still suppose that a family $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$ is a Donoho wavelets family, i.e., $\mathbf{h}$ is a symbol of an ISF of order N ($N \in \mathbb{N}$), and

$$\mathbf{g}(\omega) = \exp(-\imath\omega), \ \tilde{\mathbf{h}}(\omega) = 1, \text{ and } \tilde{\mathbf{g}}(\omega) = \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)}, \quad \omega \in \mathbb{R}.$$

Let us consider a new family $\{\mathbf{h}, \mathbf{g}^1, \tilde{\mathbf{h}}^1, \tilde{\mathbf{g}}\}$ lifted from the Donoho family as the following:

$$\begin{aligned}\mathbf{g}^1(\omega) &= \exp(-\imath\omega) - \mathbf{h}(\omega)s(2\omega),\\ \tilde{\mathbf{h}}^1(\omega) &= 1 + \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)s(2\omega)}, \quad \omega \in \mathbb{R},\end{aligned}$$

where s is a trigonometric polynomial.

Let $s(\omega) = \sum_k s_k \exp(-\imath k\omega)$, ($\omega \in \mathbb{R}$), then, we know from the theorem 4.3 that the lifted wavelet ψ^1 is

$$\psi^1(t) = 2DD_N(2t-1) - \sum_k s_k DD_N(t-k), \quad t \in \mathbb{R}.$$

Our purpose is to choose the coefficients s_k ($k \in \mathbb{Z}$) to get ψ^1 that has $2\tilde{N}$ ($N \in \mathbb{N}$) vanishing moments. From (4.8), we can write the symbol $\mathbf{h}$ of DD_N as

$$\mathbf{h}(\omega) = 1/2 + \exp(-\imath\omega)\tilde{s}(2\omega). \tag{4.9}$$

We can easily see that the coefficients of $\tilde{s}$ is half of the $2N$-points Lagrangian interpolation weights. W. Sweldens (1996 [32]) has shown the following theorem.

Theorem 4.5. *Suppose $N, \tilde{N} \in \mathbb{N}$. Let ψ be the Donoho wavelet, i.e., $\psi(\cdot) = 2DD_N(2\cdot -1)$ and $\mathbf{h}^{\tilde{N}}$ be the symbol of $DD_{\tilde{N}}$. We define $\tilde{s}^{\tilde{N}}$ as*

$$\tilde{s}^{\tilde{N}}(\omega) = \exp(\imath\omega/2)\left(\mathbf{h}^{\tilde{N}}(\omega/2) - 1/2\right).$$

If $\tilde{N} \leq N$, lifting Donoho wavelet family with

$$s(\omega) = 2\tilde{s}^{\tilde{N}}(-\omega),$$

results in the shortest wavelet with $2\tilde{N}$ vanishing moments.

Proof. For the proof of the theorem, see W. Sweldens (1996 [32]). □

We denote the lifted Donoho wavelet with $2N$ vanishing moments by Dl_N.

And we know from (4.9) and theorem 4.5 that

$$\tilde{s}(\omega) = \exp(\imath\omega/2)(\mathbf{h}(\omega/2) - 1/2) \text{ and } s(\omega) = 2\tilde{s}(-\omega).$$

Now let us consider

$$\mathbf{h}(\omega) = \frac{1}{\sqrt{2}} \sum_{l=-2N+1}^{2N-1} h_l \exp(-\imath\omega).$$

For example, if $N = 2$, then we know

$$\mathbf{h}(\omega) = \frac{1}{2}\Big(-\frac{1}{16}\exp(-\imath 3\omega) + \frac{9}{16}\exp(-\imath\omega) + 1 + \frac{9}{16}\exp(\imath\omega) - \frac{1}{16}\exp(\imath 3\omega)\Big).$$

Thus, we have

$$\begin{aligned} \tilde{s}(\omega) &= \frac{1}{2}\Big(-\frac{1}{16}\exp(-\imath\omega) + \frac{9}{16} + \frac{9}{16}\exp(\imath\omega) - \frac{1}{16}\exp(\imath 2\omega)\Big), \\ s(\omega) &= -\frac{1}{16}\exp(-\imath 2\omega) + \frac{9}{16}\exp(-\imath\omega) + \frac{9}{16} - \frac{1}{16}\exp(\imath\omega). \end{aligned}$$

Thus we know that s_k's are the 4-points Lagrangian interpolation weights and $\tilde{s}_k$'s are halves of the 4-points Lagrangian interpolation weights. This is true for the general $N \in \mathbb{N}$.

The Lifted interpolating wavelet transform

Suppose $N \in \mathbb{N}$. Let $\{DD_N, Do_N, \tilde{\phi}, \tilde{\psi}\}$ be a family of Donoho wavelets and $\{DD_N, Dl_N, \tilde{\phi}^1, \tilde{\psi}^1\}$ be a family lifted from the Donoho wavelets family. Assume that s be the corresponding lifting trigonometric polynomial,

$$s(\omega) = \sum_k s_k \exp(-\imath k\omega), \quad \text{for } \omega \in \mathbb{R},$$

where $s_k \in \mathbb{R}$. We know that these coefficients s_k ($k \in \mathbb{Z}$) are the $2N$-points symmetric Lagrangian interpolation weights (see the theorem 4.5). And let $\{\mathbf{h}, \mathbf{g}, \tilde{\mathbf{h}}, \tilde{\mathbf{g}}\}$ and $\{\mathbf{h}, \mathbf{g}^1, \tilde{\mathbf{h}}^1, \tilde{\mathbf{g}}\}$ be the corresponding family of Donoho wavelets symbols and family of lifted Donoho wavelets symbols, respectively.

Let us consider the theorem 4.3, then we know that

$$\tilde{\phi}^1(\cdot) = \sqrt{2}\sum_l \tilde{h}_l \tilde{\phi}^1(2\cdot -l) + \sum_l s_{-l}\tilde{\psi}^1(\cdot - l), \tag{4.11a}$$

$$\tilde{\psi}^1(\cdot) = \sqrt{2}\sum_l \tilde{g}_l \tilde{\phi}^1(2\cdot -l). \tag{4.11b}$$

Since

$$\mathbf{h}(\omega) = 1 \ (\omega \in \mathbb{R}),$$

we know that

$$\tilde{h}_l = \sqrt{2}\delta_{l,0} \ (l \in \mathbb{Z}).$$

And because

$$\tilde{\mathbf{g}}(\omega) = \exp(-\imath\omega)\overline{\mathbf{h}(\omega+\pi)} \text{ and } \mathbf{h}(\omega) = \frac{1}{2} + \exp(-\imath\omega)\tilde{s}(2\omega),$$

for $\omega \in \mathbb{R}$, we know that

$$\tilde{g}_l = \begin{cases} -\sqrt{2}\tilde{s}_{-l/2}, & l \text{ is even,} \\ \dfrac{1}{\sqrt{2}}\delta_{k,1}, & l \text{ is odd.} \end{cases}$$

For a function $f \in L^2(\mathbb{R})$, we define

$$\alpha_{j,k} = \langle f, \tilde{\phi}^1_{j,k} \rangle \quad \text{and} \quad \beta_{j,m} = \langle f, \tilde{\psi}^1_{j,m} \rangle.$$

Thus, from (4.11), we get

$$\alpha_{j,k} = \sqrt{2}\alpha_{j+1,2k} + \sum_l s_{-l}\beta_{j,k+l}$$

$$\beta_{j,k} = 1/\sqrt{2}\left(\alpha_{j+1,2k+1} - \sum_l 2\tilde{s}_{-l}\alpha_{j+1,2k+2l}\right).$$

This is the lifted interpolating wavelet transform. For convenience, we set

$$c_{j,k} = 2^{j/2}\alpha_{j,k} \text{ and } d_{j,k} = 2^{j/2}\beta_{j,k}.$$

Then, we have the following normalized form for the wavelet transform,

$$c_{j,k} = c_{j+1,2k} + \sum_l s_{-l}d_{j,k+l}, \tag{4.12a}$$

$$d_{j,k} = \frac{1}{2}\Big(c_{j+1,2k+1} - \sum_l 2\tilde{s}_{-l}c_{j+1,2k+2l}\Big). \tag{4.12b}$$

Now let us discuss the (4.12) in more detail. As we can see from the formula of the wavelet transform, the calculation is being done from the higher resolution level to the lower resolution level. If we assume j_1 ($\in \mathbb{N}$) to be the highest resolution level which we use to discretize the given f, i.e., f is approximated by $P_{j_1}f$, where

$$P_{j_1}f = \sum_k \alpha_{j_1,k}(DD_N)_{j_1,k}.$$

Since DD_N is interpolating, we know that $\alpha_{j_1,k} = 2^{-j_1/2}f(k/2^{j_1})$. Thus, we have

$$c_{j_1,k} = f(k/2^{j_1}).$$

4.3 AWCM for time evolution equations

In this section, we will discuss the AWCM and its application to time evolution equations. As the field values change along the time stepping, the requirements of the resolution levels of the field values should also adapt to the field profile. The adaptivity includes not only thinning the grid where the requirement of resolution level is relatively low but also adding grid points where higher resolution is probably needed in the following time step.

4.3.1 Compression of the grid points

Let f be a function in $L^2(\mathbb{R})$. And assume $j_{min}, j_{max} \in \mathbb{N}$ ($j_{min} < j_{max}$) to be the minimum resolution level and the maximum resolution level, respectively. We represent f with the lifted interpolating wavelet decomposition:

$$f = \sum_k \alpha_{j_{min},k} (DD_N)_{j_{min},k} + \sum_{j=j_{min}}^{+\infty} \sum_m \beta_{j,m} (Dl_N)_{j,m}. \tag{4.13}$$

Let V_j be the MRA generated by the scaling function DD_N and P_j be the projection of f into the subspace V_j. We consider $P_{j_{max}} f$,

$$P_{j_{max}} f = \sum_k \alpha_{j_{min},k} (DD_N)_{j_{min},k} + \sum_{j=j_{min}}^{j_{max}-1} \sum_m \beta_{j,m} (Dl_N)_{j,m}. \tag{4.14}$$

Then we have,

$$f - P_{j_{max}} f = \sum_{j=j_{max}}^{+\infty} \sum_m \beta_{j,m} (Dl_N)_{j,m}.$$

From the property of MRA, we know that

$$\lim_{j \to +\infty} \| f - P_j f \|_{L^2(\mathbb{R})} = 0.$$

Let $\zeta > 0$ be a thresholding tolerance. Then the compression $(P_{j_{max}} f)_\zeta$ of $P_{j_{max}} f$ is

$$(P_{j_{max}} f)_\zeta = \sum_k \alpha_{j_{min},k} (DD_N)_{j_{min},k} + \sum_{j=j_{min}}^{j_{max}-1} \sum_m T_\zeta(\beta_{j,m}) (Dl_N)_{j,m}, \tag{4.15}$$

where the thresholding function T_ζ is defined by

$$T_\zeta(x) := \begin{cases} x, & \text{if } |x| \geq 2^{-j/2}\zeta, \\ 0, & \text{otherwise} \end{cases}, \qquad \text{for } x \in \mathbb{R}.$$

We know that the wavelet coefficients $\beta_{j,m}$ can be calculated from the lifted interpolating wavelet transform (4.12). The thresholding criterion $|\beta_{j,m}| \geq 2^{-j/2}\zeta$ is equivalent to the normalized form $|d_{j,m}| \geq \zeta$. Note that the normalization term $2^{-j/2}$ originates from the normalization factor of $(Dl_N)_{j,m}$. We can recursively calculate all the normalized wavelet coefficients $d_{j,m} = 2^{j/2}\beta_{j,m}$ for j between j_{min} and $j_{max} - 1$.

We are interested in multi-dimensional cases, especially in 2D. In 2D case, we will consider the tensor product of scaling functions and wavelets. For $x, y \in \mathbb{R}$, let

$$\boldsymbol{\phi}(x,y) = DD_N(x) DD_N(y),$$
$$\boldsymbol{\psi}^\mu(x,y) = \begin{cases} Dl_N(x) DD_N(y), & \mu = 1, \\ DD_N(x) Dl_N(y), & \mu = 2, \\ Dl_N(x) Dl_N(y), & \mu = 3. \end{cases}$$

And we denote their translates and dilations by

$$\begin{aligned}\boldsymbol{\phi}_{j,m,n}(x,y) &= (DD_N)_{j,m}(x)(DD_N)_{j,n}(y),\\ \boldsymbol{\psi}^{\mu}_{j,m,n}(x,y) &= \begin{cases} (Dl_N)_{j,m}(x)(DD_N)_{j+1,2n}(y), & \mu = 1,\\ (DD_N)_{j+1,2m}(x)(Dl_N)_{j,n}(y), & \mu = 2,\\ (Dl_N)_{j,m}(x)(Dl_N)_{j,n}(y), & \mu = 3.\end{cases}\end{aligned}$$

For a 2D function $f \in L^2(\mathbb{R}^2)$, we have the representation formula with the lifted interpolating wavelet decomposition:

$$f(x,y) = \sum_{m,n} \alpha_{j_{min},m,n} \boldsymbol{\phi}_{j_{min},m,n}(x,y) + \sum_{\mu=1}^{3} \sum_{j=j_{min}}^{+\infty} \sum_{m,n} \beta^{\mu}_{j,m,n} \boldsymbol{\psi}^{\mu}_{j,m,n}(x,y).$$

First, we state the following facts on tensor products without proof.

- Let $\mathbf{H}_1$ and $\mathbf{H}_2$ be two Hilbert spaces. In particular, here we only deal with Hilbert spaces of 1D functions. Define a tensor product $\mathbf{H}$ of $\mathbf{H}_1$ and $\mathbf{H}_2$ as

  $$\overline{span\{f_1 \otimes f_2 \mid f_1 \in \mathbf{H}_1, f_2 \in \mathbf{H}_2\}},$$

  where $f_1 \otimes f_2(\cdot,*) := f_1(\cdot)f_2(*)$. We denote $\mathbf{H}$ by $\mathbf{H}_1 \otimes \mathbf{H}_2$. Then $\mathbf{H}$ is a Hilbert space, with the inner product:

  $$\langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle_{\mathbf{H}} = \langle f_1, g_1 \rangle_{\mathbf{H}_1} \langle f_2, g_2 \rangle_{\mathbf{H}_2},$$

  where $f_1, g_1 \in \mathbf{H}_1$ and $f_2, g_2 \in \mathbf{H}_2$. Furthermore, if $\{e^1_n\}_{n\in\mathbb{N}}$ and $\{e^2_n\}_{n\in\mathbb{N}}$ are two Riesz bases of $\mathbf{H}_1$ and $\mathbf{H}_2$ respectively, then $\{e^1_n \otimes e^2_m\}_{n,m\in\mathbb{N}}$ is a Riesz basis of $\mathbf{H}$.
- $L^2(\mathbb{R}^2) = L^2(\mathbb{R}) \otimes L^2(\mathbb{R})$.
- Let $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA of $L^2(\mathbb{R})$ generated by DD_N. Define $V^2_j = V_j \otimes V_j$. Then $\{V^2_j\}$ is an MRA of $L^2(\mathbb{R}^2)$ generated by $\boldsymbol{\phi}_N$.

Now we use $\mathbf{P}_j f$ to denote a projection of f into V^2_j. Then, we have

$$\mathbf{P}_{j_{max}} f = \sum_{m,n} \alpha_{j_{min},m,n} \boldsymbol{\phi}_{j_{min},m,n} + \sum_{\mu=1}^{3} \sum_{j=j_{min}}^{j_{max}-1} \sum_{m,n} \beta^{\mu}_{j,m,n} \boldsymbol{\psi}^{\mu}_{j,m,n}. \tag{4.16}$$

Then the compression $(\mathbf{P}_{j_{max}} f)_\zeta$ of $\mathbf{P}_{j_{max}} f$ is

$$(\mathbf{P}_{j_{max}} f)_\zeta = \sum_{m,n} \alpha_{j_{min},m,n} \boldsymbol{\phi}_{j_{min},m,n} + \sum_{\mu=1}^{3} \sum_{j=j_{min}}^{j_{max}-1} \sum_{m,n} T^{\mu}_{\zeta}(\beta^{\mu}_{j,m,n}) \boldsymbol{\psi}^{\mu}_{j,m,n}, \tag{4.17}$$

where the thresholding function T^{μ}_{ζ} is defined by

$$T^{\mu}_{\zeta}(x) = \begin{cases} x, & \text{if } |x| \geq 2^{-j-1/2}\zeta, \\ 0, & \text{otherwise}\end{cases}, \qquad \text{for } \mu = 1, 2,\ x \in \mathbb{R}$$

and

$$T_\zeta^3(x) = \left\{ \begin{array}{ll} x, & \text{if } |x| \geq 2^{-j}\zeta, \\ 0, & \text{otherwise} \end{array} \right. , \quad \text{for } x \in \mathbb{R}.$$

The terms $2^{-j-1/2}$ and 2^{-j} before ζ are normalized factors, which come from the normalization factors of $\boldsymbol{\psi}^\mu_{j,m,n}$ (See the definition of $d^\mu_{j,m,n}$ below).

By discarding the wavelet coefficients which are negligible, we can compress the representation of a function. We will consider the dual scaling function and wavelet in order to calculate these coefficients. Let $\tilde{\phi}$ and $\tilde{\psi}$ be the 1D dual scaling function and dual wavelet of the lifted interpolating wavelet family. The 2D dual scaling function $\tilde{\boldsymbol{\phi}}$ and dual wavelet $\tilde{\boldsymbol{\psi}}^\mu$ $(\mu = 1, 2, 3)$ are defined by their tensor products:

$$\begin{aligned} \tilde{\boldsymbol{\phi}}(x, y) &= \tilde{\phi}(x)\tilde{\phi}(y), \\ \tilde{\boldsymbol{\psi}}^\mu(x, y) &= \left\{ \begin{array}{ll} \tilde{\psi}(x)\tilde{\phi}(y), & \mu = 1, \\ \tilde{\phi}(x)\tilde{\psi}(y), & \mu = 2, \\ \tilde{\psi}(x)\tilde{\psi}(y), & \mu = 3. \end{array} \right. \end{aligned}$$

Since we know that

$$\tilde{g}(\omega) = \exp(-\imath\omega)\overline{\mathbf{h}(\omega + \pi)} \text{ and } \mathbf{h}(\omega) = \frac{1}{2} + \exp(-\imath\omega)\tilde{s}(2\omega), \quad \omega \in \mathbb{R},$$

the equation (4.11) becomes:

$$\begin{aligned} \tilde{\phi}(\cdot) &= 2\tilde{\phi}(2\cdot) + \sum_l s_{-l}\tilde{\psi}(\cdot - l), \\ \tilde{\psi}(\cdot) &= \tilde{\phi}(2 \cdot -1) - \sum_l 2\tilde{s}_{-l}\tilde{\phi}(2 \cdot -2l). \end{aligned}$$

We define the translates and dilations of dual functions as

$$\begin{aligned} \tilde{\boldsymbol{\phi}}_{j,m,n}(x, y) &= \tilde{\phi}_{j,m}(x)\tilde{\phi}_{j,n}(y), \\ \tilde{\boldsymbol{\psi}}^\mu_{j,m,n}(x, y) &= \left\{ \begin{array}{ll} \tilde{\psi}_{j,m}(x)\tilde{\phi}_{j+1,2n}(y), & \mu = 1, \\ \tilde{\phi}_{j+1,2m}(x)\tilde{\psi}_{j,n}(y), & \mu = 2, \\ \tilde{\psi}_{j,m}(x)\tilde{\psi}_{j,n}(y), & \mu = 3. \end{array} \right. \end{aligned}$$

Thus,

$$\begin{aligned}
\tilde{\boldsymbol{\phi}}_{j,m,n}(x,y) =& \tilde{\phi}_{j,m}(x)\tilde{\phi}_{j,n}(y)\\
=& 2^{j}\tilde{\phi}(2^{j}x-m)\tilde{\phi}(2^{j}y-n)\\
=& 2^{j}\Big(2\tilde{\phi}(2^{j+1}x-2m)+\sum_{l}s_{-l}\tilde{\psi}(2^{j}x-m-l)\Big)\Big(2\tilde{\phi}(2^{j+1}y-2n)\\
&+\sum_{l'}s_{-l'}\tilde{\psi}(2^{j}y-n-l')\Big)\\
=& 2^{j}\Big(4\tilde{\phi}(2^{j+1}x-2m)\tilde{\phi}(2^{j+1}y-2n)\\
&+2\sum_{l}s_{-l}\tilde{\psi}(2^{j}x-m-l)\tilde{\phi}(2^{j+1}y-2n)\\
&+2\sum_{l'}s_{-l'}\tilde{\phi}(2^{j+1}x-2m)\tilde{\psi}(2^{j}y-n-l')\\
&+\sum_{l}\sum_{l'}s_{-l}s_{-l'}\tilde{\psi}(2^{j}x-m-l)\tilde{\psi}(2^{j}y-n-l')\Big)\\
=& 2\tilde{\boldsymbol{\phi}}_{j+1,2m,2n}(x,y)+\sqrt{2}\sum_{l}s_{-l}\tilde{\boldsymbol{\psi}}^{1}_{j,m+l,n}(x,y)\\
&+\sqrt{2}\sum_{-l'}s_{-l'}\tilde{\boldsymbol{\psi}}^{2}_{j,m,n+l'}(x,y)+\sum_{l}\sum_{l'}s_{-l}s_{-l'}\tilde{\boldsymbol{\psi}}^{3}_{j,m+l,n+l'}(x,y).
\end{aligned}$$

$$\begin{aligned}
\tilde{\boldsymbol{\psi}}^{1}_{j,m,n}(x,y) =& \tilde{\psi}_{j,m}(x)\tilde{\phi}_{j+1,2n}(y)\\
=& 2^{j/2}\tilde{\psi}(2^{j}x-m)\tilde{\phi}_{j+1,2n}(y)\\
=& 2^{j/2}\Big(\tilde{\phi}(2^{j+1}x-2m-1)-\sum_{l}2\tilde{s}_{-l}\tilde{\phi}(2^{j+1}x-2m-2l)\Big)\tilde{\phi}_{j+1,2n}(y)\\
=& \frac{1}{\sqrt{2}}\Big(\tilde{\phi}_{j+1,2m+1}(x)\tilde{\phi}_{j+1,2n}(y)-\sum_{l}2\tilde{s}_{-l}\tilde{\phi}_{j+1,2m+2l}(x)\tilde{\phi}_{j+1,2n}(y)\Big)\\
=& \frac{1}{\sqrt{2}}\Big(\tilde{\boldsymbol{\phi}}_{j+1,2m+1,2n}(x,y)-\sum_{l}2\tilde{s}_{-l}\tilde{\boldsymbol{\phi}}_{j+1,2m+2l,2n}(x,y)\Big).
\end{aligned}$$

In the same way, we obtain

$$\tilde{\boldsymbol{\psi}}^{2}_{j,m,n}(x,y)=\frac{1}{\sqrt{2}}\Big(\tilde{\boldsymbol{\phi}}_{j+1,2m,2n+1}(x,y)-\sum_{l}2\tilde{s}_{-l}\tilde{\boldsymbol{\phi}}_{j+1,2m,2n+2l}(x,y)\Big).$$

and

$$\begin{aligned}
\tilde{\boldsymbol{\psi}}^{3}_{j,m,n} =& \frac{1}{2}\Big(\tilde{\boldsymbol{\phi}}_{j+1,2m+1,2n+1}-\sum_{l}2\tilde{s}_{-l}\tilde{\boldsymbol{\phi}}_{j+1,2m+2l,2n+1}\\
&-\sum_{l'}2\tilde{s}_{-l'}\tilde{\boldsymbol{\phi}}_{j+1,2m+1,2n+2l'}+\sum_{l}\sum_{l'}(2\tilde{s}_{-l})(2\tilde{s}_{-l'})\tilde{\boldsymbol{\phi}}_{j+1,2m+2l,2n+2l'}\Big).
\end{aligned}$$

Now we come to the calculations of the coefficients $\alpha_{j,m,n}$ and $\beta^{\mu}_{j,m,n}$ of the wavelet decomposition (4.16). We know that

$$\alpha_{j,m,n} = \langle f, \tilde{\boldsymbol{\phi}}_{j,m,n}\rangle \text{ and } \beta^{\mu}_{j,m,n} = \langle f, \tilde{\boldsymbol{\psi}}^{\mu}_{j,m,n}\rangle.$$

Therefore, we have the following 2D fast wavelet transform:

$$\begin{aligned}
\alpha_{j,m,n} =& 2\alpha_{j+1,2m,2n} + \sqrt{2}\sum_{l} s_{-l}\beta^1_{j,m+l,n} + \sqrt{2}\sum_{l'} s_{-l'}\beta^2_{j,m,n+l'} + \sum_{l}\sum_{l'} s_{-l}s_{-l'}\beta^3_{j,m+l,n+l'},\\
\beta^1_{j,m,n} =& \frac{1}{\sqrt{2}}\Big(\alpha_{j+1,2m+1,2n} - \sum_{l} 2\tilde{s}_{-l}\alpha_{j+1,2m+2l,2n}\Big),\\
\beta^2_{j,m,n} =& \frac{1}{\sqrt{2}}\Big(\alpha_{j+1,2m,2n+1} - \sum_{l} 2\tilde{s}_{-l}\alpha_{j+1,2m,2n+2l}\Big),\\
\beta^3_{j,m,n} =& \frac{1}{2}\Big(\alpha_{j+1,2m+1,2n+1} - \sum_{l} 2\tilde{s}_{-l}\alpha_{j+1,2m+2l,2n+1} - \sum_{l'} 2\tilde{s}_{-l'}\alpha_{j+1,2m+1,2n+2l'}\\
&+ \sum_{l}\sum_{l'}(2\tilde{s}_{-l})(2\tilde{s}_{-l'})\alpha_{j+1,2m+2l,2n+2l'}\Big).
\end{aligned}$$

To get a normalized form of the fast wavelet transform, we set

$$c_{j,m,n} = 2^j\alpha_{j,m,n},\ d^1_{j,m,n} = 2^{j+1/2}\beta^1_{j,m,n},\ d^2_{j,m,n} = 2^{j+1/2}\beta^2_{j,m,n} \text{ and } d^3_{j,m,n} = 2^j\beta^3_{j,m,n}.$$

Then, we have the 2D lifted interpolating wavelet transform in the normalized form:

$$d^1_{j,m,n} = \frac{1}{2}\Big(c_{j+1,2m+1,2n} - \sum_{l} 2\tilde{s}_{-l}c_{j+1,2m+2l,2n}\Big), \tag{4.18a}$$

$$d^2_{j,m,n} = \frac{1}{2}\Big(c_{j+1,2m,2n+1} - \sum_{l} 2\tilde{s}_{-l}c_{j+1,2m,2n+2l}\Big), \tag{4.18b}$$

$$\begin{aligned}
d^3_{j,m,n} =& \frac{1}{4}\Big(c_{j+1,2m+1,2n+1} - \sum_{l} 2\tilde{s}_{-l}c_{j+1,2m+2l,2n+1} - \sum_{l'} 2\tilde{s}_{-l'}c_{j+1,2m+1,2n+2l'}\\
&+ \sum_{l}\sum_{l'}(2\tilde{s}_{-l})(2\tilde{s}_{-l'})c_{j+1,2m+2l,2n+2l'}\Big).
\end{aligned} \tag{4.18c}$$

$$c_{j,m,n} = c_{j+1,2m,2n} + \sum_{l} s_{-l}d^1_{j,m+l,n} + \sum_{l'} s_{-l'}d^2_{j,m,n+l'} + \sum_{l}\sum_{l'} s_{-l}s_{-l'}d^3_{j,m+l,n+l'}, \tag{4.18d}$$

It is clear that the criterion of thresholding is simplified into $|d^{\mu}_{j,m,n}| < \zeta$.

And the inverse wavelet transform is

$$c_{j+1,2m,2n} = c_{j,m,n} - \sum_l s_{-l} d^1_{j,m+l,n} + \sum_{l'} s_{-l'} d^2_{j,m,n+l'} + \sum_l \sum_{l'} s_{-l} s_{-l'} d^3_{j,m+l,n+l'}, \tag{4.19a}$$

$$c_{j+1,2m+1,2n} = 2d^1_{j,m,n} + \sum_l 2\tilde{s}_{-l} c_{j+1,2m+2l,2n} \tag{4.19b}$$

$$c_{j+1,2m,2n+1} = 2d^2_{j,m,n} + \sum_l 2\tilde{s}_{-l} c_{j+1,2m,2n+2l} \tag{4.19c}$$

$$\begin{aligned} c_{j+1,2m+1,2n+1} =& 4d^3_{j,m,n} + \sum_l 2\tilde{s}_{-l} c_{j+1,2m+2l,2n+1} + \sum_{l'} 2\tilde{s}_{-l'} c_{j+1,2m+1,2n+2l'} \\ & - \sum_l \sum_{l'} (2\tilde{s}_{-l})(2\tilde{s}_{-l'}) c_{j+1,2m+2l,2n+2l'} \end{aligned} \tag{4.19d}$$

Now we discuss the relation between these coefficients and the numerical grid points. Let $j_{min}, j_{max} \in \mathbb{Z}$ $(j_{min} < j_{max})$. For any integer j between j_{min} and j_{max}, we define the coordinates of mesh points in the resolution level j as

$$x_{j,m} = \frac{m}{2^j} \quad \text{and} \quad y_{j,n} = \frac{n}{2^j}, \quad \text{for} \quad m, n \in \mathbb{Z}.$$

We define

$$\mathcal{K}_j := \{(x_{j,m}, y_{j,n}) \mid m, n \in \mathbb{Z}\}.$$

Then, we know that $\mathcal{K}_j$'s have the nested relation, i.e.,

$$\mathcal{K}_j \subset \mathcal{K}_{j+1}, \quad \text{for} \quad j = j_{min}, j_{min}+1, \cdots, j_{max}-1.$$

It is easy to check that $x_{j,m} = x_{j+1,2m}$ and $y_{j,n} = y_{j+1,2n}$. We also define

$$\mathcal{M}^\mu_j := \begin{cases} \{(x_{j+1,2m+1}, y_{j+1,2n}) \mid m, n \in \mathbb{Z}\}, & \text{if } \mu = 1, \\ \{(x_{j+1,2m}, y_{j+1,2n+1}) \mid m, n \in \mathbb{Z}\}, & \text{if } \mu = 2, \\ \{(x_{j+1,2m+1}, y_{j+1,2n+1}) \mid m, n \in \mathbb{Z}\}, & \text{if } \mu = 3. \end{cases}$$

And we set

$$\mathcal{M}_j := \mathcal{M}^1_j \cup \mathcal{M}^2_j \cup \mathcal{M}^3_j,$$

then, we have

$$\mathcal{M}_j = \mathcal{K}_{j+1} \setminus \mathcal{K}_j.$$

We can also represent $\mathbf{P}_{j_{max}} f = \sum_{m,n} \alpha_{j_{max},m,n} \boldsymbol{\phi}_{j_{max},m,n}$. Thus, we know that

$$\alpha_{j_{max},m,n} = 2^{-j_{max}} \mathbf{P}_{j_{max}} f\Big(\frac{m}{2^{j_{max}}}, \frac{n}{2^{j_{max}}}\Big),$$

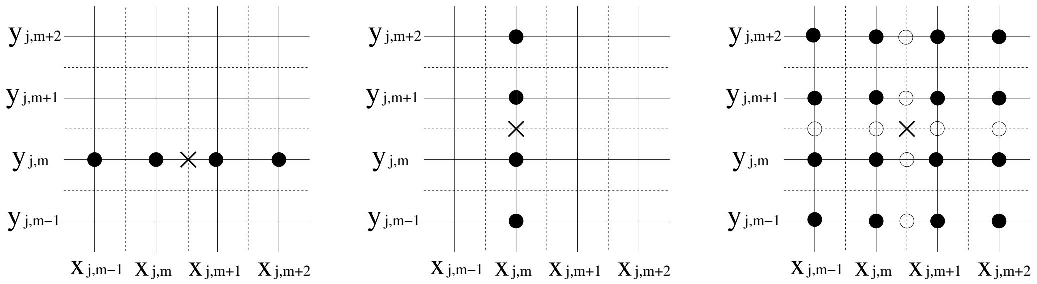
or in the normalized form,

$$c_{j_{max},m,n} = \mathbf{P}_{j_{max}} f\Big(\frac{m}{2^{j_{max}}}, \frac{n}{2^{j_{max}}}\Big).$$

We start the fast wavelet transform with the values $c_{j_{max},m,n}$ of the highest resolution level j_{max} to calculate $d^{\mu}_{j_{max}-1,m,n}$'s and $c_{j_{max}-1,m,n}$, and then we go on with the values $c_{j_{max}-1,m,n}$ and repeatedly compute coefficients $c_{j,m,n}$'s and $d^{\mu}_{j,m,n}$'s until we reach the lowest resolution level j_{min}. We have the following one to one correspondence between these coefficients and grid points.

$$\begin{aligned}
c_{j_{min},m,n} &:(x_{j_{min},m}, y_{j_{min},n}), \quad m,n \in \mathbb{Z} \\
d^1_{j,m,n} &:(x_{j+1,2m+1}, y_{j+1,2n}), \quad m,n \in \mathbb{Z},\ j = j_{min}, j_{min}+1, \cdots, j_{max}-1 \\
d^2_{j,m,n} &:(x_{j+1,2m}, y_{j+1,2n+1}), \quad m,n \in \mathbb{Z},\ j = j_{min}, j_{min}+1, \cdots, j_{max}-1 \\
d^3_{j,m,n} &:(x_{j+1,2m+1}, y_{j+1,2n+1}), \quad m,n \in \mathbb{Z},\ j = j_{min}, j_{min}+1, \cdots, j_{max}-1
\end{aligned}$$

By discarding the grid points whose absolute values of the corresponding wavelet coefficients are less than the given thresholding tolerance $\varepsilon > 0$, we have the compression of the grid points. However, we do not perform the fast wavelet transform only one time. When we solve a time evolution equation, we perform the fast wavelet transform every time step. Suppose a wavelet coefficient $d^{\mu}_{j,m,n}$ survived after the compression at the time step $n\Delta t$. Then, at the next time step $(n+1)\Delta t$, in order to calculate $d^{\mu}_{j,m,n}$, we need neighbor points of the point which corresponds to $d^{\mu}_{j,m,n}$ (See Figure 4.1).



(a) ×: point corresponds to $d^1_{j,m,n}$; •: neighbor points needed to calculate $d^1_{j,m,n}$.

(b) ×: point corresponds to $d^2_{j,m,n}$; •: neighbor points needed to calculate $d^2_{j,m,n}$.

(c) ×: point corresponds to $d^3_{j,m,n}$; •, ∘: neighbor points needed to calculate $d^3_{j,m,n}$.

Figure 4.1: Descriptions of the neighbor points needed to calculate wavelet coefficients $d^{\mu}_{j,m,n}$ with the order $N = \tilde{N} = 2$.

The process of adding these neighbor points needed to calculate wavelet coefficients is called *reconstruction check*. The efficiency of the wavelet transform depends on the number of the finest grid points at the beginning; however, after the first compression, it depends on the number of compressed or adaptive grid points. Thus, the profile of the field values itself determines the speed of the computation.

Of course, we cannot only discard the *unimportant* grid points in every time step. Since some part of the adaptive grid which is unimportant in some time step does not necessarily stay unimportant in the next time step, we ought to consider artificially making some *potential part* of the grid finer. The adaptivity of numerical grid does not only mean compression, but also includes the process of *extension*. We will discuss this extension of the so called *adjacent zone* in the next subsection.

4.3.2 Adding adjacent zone

Here we want to consider the concept of the *adjacent zone* with the example of the time domain Maxwell's equations. When solving the time domain Maxwell's equations numerically, we must choose the smallest time step according to the numerical stability condition. From the numerical stability condition we easily know that wave information at one position does not travel more than one spatial cell in one time step. So it is reasonable for us to suppose that only the wavelet coefficients which are not *far away* from the significant wavelet coefficients in some time step would be possible to become *active* in the following time step, see for example, J. Liandrat and P. Tchamitchian 1990 [22] and O. Vasilyev 2000 [40], 2003 [39].

We will follow Vasilyev's way of description about the adjacent zone. For 2D case, we assume a wavelet coefficient $d^{\mu}_{j,m,n}$ *survived* after the compression of the grid points in the current time step. We assume $(x_{j+1,k_1}, y_{j+1,k_2})$ to be the point corresponding to $d^{\mu}_{j,m,n}$. Then we require that the wavelet coefficient $d^{\mu}_{j',m',n'}$ located at the point $(x_{j'+1,k'_1}, y_{j'+1,k'_2})$ should belong to the adjacent zone if

$$|j'-j| \leq L, \quad |2^{j'-j}k_1 - k'_1| \leq M, \quad |2^{j'-j}k_2 - k'_2| \leq M, \quad \text{for } L, M \in \mathbb{N},$$

where L explains the range of the resolution levels that should be added around an existing wavelet coefficient and M is the width of the adjacent zone. We choose $L = M = 1$. In other words, if a point $P \in \mathcal{M}_j$ $(j_{min} \leq j \leq j_{max} - 1)$ is in the adaptive grid, we add eight nearest points of P in $\mathcal{K}_{j+1}$. Furthermore, if $j < j_{max} - 1$, we add additional eight nearest points of P in $\mathcal{K}_{j+2}$. See Figure 4.3.2.

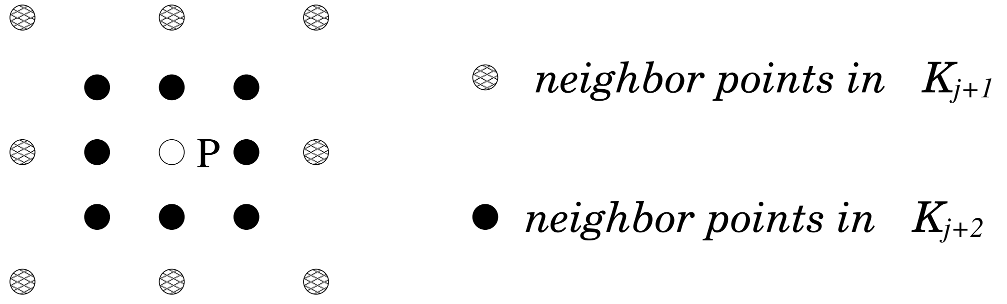


Figure 4.2: Description of adjacent zone of a point P in $\mathcal{M}_j$.

4.3.3 Approximation of spatial derivatives on dynamic grid

We continue to discuss the approximation of spatial derivatives of a function on an adaptive grid. For a 2D function f in $L^2(\mathbb{R}^2)$. Let us consider the lifted interpolating wavelet decomposition of f. Assume j_{min} and j_{max} to be the lowest resolution level and the highest resolution level respectively. For a given tolerance $\zeta > 0$, we compress the projection $\mathbf{P}_{j_{max}}f$ of f as the following,

$$(\mathbf{P}_{j_{max}}f)_{\zeta} = \sum_{m,n} \alpha_{j_{min},m,n}\boldsymbol{\phi}_{j_{min},m,n} + \sum_{\mu=1}^{3}\sum_{j=j_{min}}^{j_{max}-1}\sum_{m,n} T^{\mu}_{\zeta}(\beta^{\mu}_{j,m,n})\boldsymbol{\psi}^{\mu}_{j,m,n}.$$

We will use derivatives of $(\mathbf{P}_{j_{max}}f)_\zeta$ to approximate those of f. Before we calculate the derivatives of $(\mathbf{P}_{j_{max}}f)_\zeta$ at each point in an adaptive grid, we ought to first determine the *density level* of the point in the adaptive grid. The density level of a point in an adaptive grid is the maximum of the x-level and the z-level of that point.

We only discuss the x-level since the z-level is similar. For each point $Q = (x_0, z_0)$ in an adaptive grid $\mathcal{G}$, we define the x-level of Q in $\mathcal{G}$ according to a point $Q' = (x_1, z_0)$ in $\mathcal{G}$ nearest to Q. The x-level $Levelx$ of Q in $\mathcal{G}$ is

$$Levelx := j_{max} - \log_2(dist(Q, Q')/\Delta x), \tag{4.20}$$

where Δx is the smallest computational mesh size along x axis and $dist(Q, Q') = |x_1 - x_0|$. If $dist(Q, Q') = \Delta x$, that means the level $Levelx$ of Q is the maximum, j_{max}, and if $dist(Q, Q') = 2\Delta x$, then the $Levelx$ of Q is $j_{max} - 1$, and so on. The z-level of a point in an adaptive grid is defined in the same way as x-level. Then the density level of a point in an adaptive grid is defined as the maximum value of x-level and z-level of the point in the adaptive grid. See Figure 4.3.

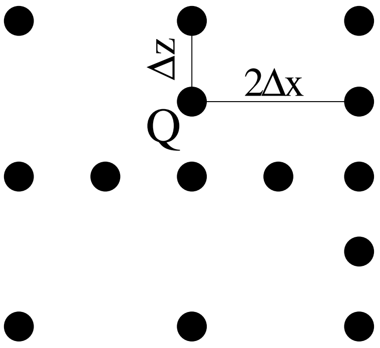


Figure 4.3: Description of the density level of a point Q in an adaptive grid: the x-level of Q is $j_{max} - 1$ and the z-level of Q is j_{max}, thus, the density level of Q in the adaptive grid is j_{max}.

Now we continue to discuss the derivative calculations. Suppose j_0 to be the density level of Q in the adaptive grid $\mathcal{G}$. Then, we can approximate $(\mathbf{P}_{j_{max}}f)_\zeta$ by $\mathbf{P}_{j_0}f$ locally at some neighborhood Ω_0 of Q.

$$(\mathbf{P}_{j_0}f)(x, y) = \sum_{m,n} \alpha_{j_0,m,n}\boldsymbol{\phi}_{j_0,m,n}(x, y), \quad (x, y) \in \Omega_0 \tag{4.21}$$

We differentiate $\mathbf{P}_{j_0}f(x, y)$ to approximate x-derivative of f at Q. If some points in the sum (4.21) are not present in $\mathcal{G}$, we interpolate the values at those points using values of the coarser levels with inverse wavelet transform. We know that

$$\alpha_{j_0,m,n} = 2^{-j_0}(\mathbf{P}_{j_0}f)\Big(\frac{m}{2^{j_0}}, \frac{n}{2^{j_0}}\Big), \quad \text{for } m, n \in \mathbb{Z}.$$

Thus, we have

$$(\mathbf{P}_{j_0}f)(x, y) = \sum_{m,n}(\mathbf{P}_{j_0}f)\Big(\frac{m}{2^{j_0}}, \frac{n}{2^{j_0}}\Big)DD_N\big(2^{j_0}x - m\big)DD_N\big(2^{j_0}y - n\big), \quad (x, y) \in \Omega_0 \tag{4.22}$$

When we differentiate both sides of (4.22), we need the differentiation filters which have been computed in the previous chapter, see the Table 3.1. For example,

$$\frac{(\partial \mathbf{P}_{j_0} f)}{\partial x}(x,y) = \sum_{m,n} (\mathbf{P}_{j_0} f)\Big(\frac{m}{2^{j_0}}, \frac{n}{2^{j_0}}\Big) \frac{dDD_N\big(2^{j_0}x - m\big)}{dx} DD_N\big(2^{j_0}y - n\big), \quad (x,y) \in \Omega_0. \tag{4.23}$$

The R.H.S. of (4.23) is a finite sum of products of separated functions of x and y. And since the density level of Q is j_0, there exist $m', n' \in \mathbb{Z}$ such that $Q = \Big(\frac{m'}{2^{j_0}}, \frac{n'}{2^{j_0}}\Big)$. It is easy to see that

$$\begin{aligned}\frac{(\partial \mathbf{P}_{j_0} f)}{\partial x}\Big(\frac{m'}{2^{j_0}}, \frac{n'}{2^{j_0}}\Big) =& \sum_{m,n} (\mathbf{P}_{j_0} f)\Big(\frac{m}{2^{j_0}}, \frac{n}{2^{j_0}}\Big) \frac{dDD_N\big(m' - m\big)}{dx} DD_N\big(n' - n\big).\\ =& 2^{j_0} \sum_{m} (\mathbf{P}_{j_0} f)\Big(\frac{m}{2^{j_0}}, \frac{n'}{2^{j_0}}\Big) (DD_N)'\big(m' - m\big).\end{aligned} \tag{4.24}$$

Similarly, we obtain

$$\frac{(\partial \mathbf{P}_{j_0} f)}{\partial z}\Big(\frac{m'}{2^{j_0}}, \frac{n'}{2^{j_0}}\Big) = 2^{j_0} \sum_{n} (\mathbf{P}_{j_0} f)\Big(\frac{m'}{2^{j_0}}, \frac{n}{2^{j_0}}\Big) (DD_N)'\big(n' - n\big). \tag{4.25}$$

4.3.4 General steps of the algorithm

We talk about the general steps of the AWCM with the example of TM$_y$ mode equations. In TM$_y$ mode equations, we have three unknown functions $\mathcal{E}_y$, $\mathcal{H}_x$ and $\mathcal{H}_z$. These are time dependent functions defined on xz plane in space, for example, $\mathcal{E}_y(x,z,t)$. Assume that we solve the equations on a square domain $\Omega = \big[-\frac{L}{2}, \frac{L}{2}\big] \times \big[-\frac{L}{2}, \frac{L}{2}\big]$. Let j_{min} and j_{max} be the coarsest and highest resolution levels respectively. We discretize the domain Ω by $2^{j_{max}} \times 2^{j_{max}}$ small square subdomains. We define $x_{j,m} := \dfrac{mL}{2^j}$ and $y_{j,n} := \dfrac{nL}{2^j}$, for $j_{min} \le j \le j_{max}$, $m, n = 0, 1, \cdots, 2^j$. And we define the nested gird sets $\mathcal{K}_j$ and $\mathcal{M}_j^\mu$ as

$$\mathcal{K}_j := \{(x_{j,m}, y_{j,n}) \mid m, n = 0, 1, \cdots, 2^j.\}$$

for $j_{min} \le j \le j_{max}$, and

$$\mathcal{M}_j^\mu := \begin{cases} \{(x_{j+1,2m+1}, y_{j+1,2n}) \mid m = 0, 1, \cdots, 2^j - 1, n = 0, 1, \cdots, 2^j\}, & \text{if } \mu = 1,\\ \{(x_{j+1,2m}, y_{j+1,2n+1}) \mid m = 0, 1, \cdots, 2^j, n = 0, 1, \cdots, 2^j - 1\}, & \text{if } \mu = 2,\\ \{(x_{j+1,2m+1}, y_{j+1,2n+1}) \mid m, n = 0, 1, \cdots, 2^j - 1\}, & \text{if } \mu = 3.\end{cases}$$

for $j \in \mathbb{N}$ ($j_{min} \le j \le j_{max} - 1$).

In AWCM, we use unstaggered collocated scheme instead of Yee's scheme, i.e., all the fields are stored at the same position. At the beginning, all the fields are discretized on $\mathcal{K}_{j_{max}}$. For electric field component $\mathcal{E}_y$, we consider the ISF representation of it. Note that we have the scaling factor $1/L$ in the basis function decomposition.

$$\mathbf{P}_{j_{max}} \mathcal{E}_y\Big(\frac{x}{L}, \frac{z}{L}, t\Big) = \sum_{m,n} \alpha_{j_{max},m,n}(t) \boldsymbol{\phi}_{j_{max},m,n}\Big(\frac{x}{L}, \frac{z}{L}\Big).$$

From the interpolating property of ISF, we get all the scaling coefficients $\alpha_{j_{max},m,n}$.

$$\alpha_{j_{max},m,n}(t) = 2^{-j_{max}}\mathcal{E}_y\Big(\frac{x_{j_{max},m}}{L}, \frac{z_{j_{max},n}}{L}, t\Big),$$

or in the normalized form

$$c_{j_{max},m,n}(t) = \mathcal{E}_y\Big(\frac{x_{j_{max},m}}{L}, \frac{z_{j_{max},n}}{L}, t\Big),$$

Starting with these coefficients, we perform the lifted interpolating wavelet transform (4.18) to get all the coefficients in the wavelet decomposition of the field,

$$\mathbf{P}_{j_{max}}\mathcal{E}_y\Big(\frac{x}{L}, \frac{z}{L}, t\Big) = \sum_{m,n}\alpha_{j_{min},m,n}(t)\boldsymbol{\phi}_{j_{min},m,n}\Big(\frac{x}{L}, \frac{z}{L}\Big) + \sum_{\mu=1}^{3}\sum_{j=j_{min}}^{j_1-1}\sum_{m,n}\beta^{\mu}_{j,m,n}(t)\boldsymbol{\psi}^{\mu}_{j,m,n}\Big(\frac{x}{L}, \frac{z}{L}\Big).$$

In practical coding, instead of using these original coefficients $\alpha_{j,m,n}$ and $\beta^{\mu}_{j,m,n}$, we use the normalized coefficients $c_{j,m,n}$'s and $d^{\mu}_{j,m,n}$'s.

Let us consider (1.26). For example, we update $\mathcal{B}_x$ and $\mathcal{H}_x$ on an adaptive grid $\mathcal{G}$ in the following way. For a point $Q \in \mathcal{G}$, we use $\mathcal{A}|^k_Q$ to denote discretized value of $\mathcal{A}$ at Q in the time step $k\Delta t$. Assume $j(Q)$ to be the density level of Q in the grid $\mathcal{G}$. We can represent Q as $(x_{j(Q),m'}, y_{j(Q),n'})$ for some $m', n' \in \mathbb{Z}$. Then we have the following updating equations,

$$\mathcal{B}_x|^{k+1/2}_Q = \mathcal{B}_x|^{k-1/2}_Q + \Delta t\frac{2^{j(Q)}}{L}\sum_n \mathcal{E}_y|^k_{(x_{j(Q),m'},y_{j(Q),n})}(DD_N)'(n'-n), \tag{4.26a}$$

$$\mathcal{H}_x|^{k+1/2}_Q = \frac{1-\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{H}_x|^{k-1/2}_Q + \frac{1}{\mu_0}\left(\frac{1+\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{B}_x|^{k+1/2}_Q - \frac{1-\dfrac{\sigma_x\Delta t}{2\varepsilon_0}}{1+\dfrac{\sigma_z\Delta t}{2\varepsilon_0}}\mathcal{B}_x|^{k-1/2}_Q\right). \tag{4.26b}$$

AWCM algorithm for solving TM$_y$ mode equations

Here we want to discuss the AWCM algorithm for solving TM$_y$ mode equations.

First, we initialize all the necessary global variables which will be called by other subroutines. For example, the maximum and minimum resolution levels j_{max} and j_{min}, the order N of DD_N and the number $\tilde{N}$ (normally equal to N) of the vanishing moments of the lifted wavelet $Dl_{\tilde{N}}$, and the geometric parameters of the computational domain, such as length L of the computational domain, the smallest mesh sizes Δx and Δz, and the updating coefficients in PML region, etc.

We will use 2D arrays of real numbers for storing the field values such as $\mathcal{E}_y$, $\mathcal{H}_x$, $\mathcal{H}_z$, $\mathcal{D}_y$, $\mathcal{B}_x$ and $\mathcal{B}_z$. The initial conditions of these field values are included in the process of the initialization. For the storage of an adaptive grid, we will use a 2D array of booleans which is called a *grid mask*, or simply a *mask*. If the value of an entry of a mask is true or 1, that means the corresponding grid point is included in the adaptive grid; otherwise, the grid point is not in the adaptive grid. Moreover, by forcing the value of an entry of a mask into 1, we mean the *inclusion* of the corresponding point into the adaptive grid, or by forcing the value of an entry of a mask into 0, we mean the corresponding point is *removed* from the adaptive grid. Initially, we start with a full mask grid, i.e., all entries have values 1.

The Algorithm 1 is a pseudo-code of the main function awcm2d_main(). After the initialization awcm2d_initialize(), it goes on with time stepping repeatedly by a "for" loop, which contains, awcm2d_adaptive(), awcm2d_update() and awcm2d_print(). In the awcm2d_adaptive(), the grid will be adapted by performing wavelet transforms to the field value $\mathcal{E}_y$. In the awcm2d_update(), all the fields will be updated on some adaptive grids. And then we output the calculated field data into files for visualization in awcm2d_print().

Algorithm 1: awcm2d_main for TM$_y$ mode equations

```
# initialization
awcm2d_initialize()
# ————————————————————————————————
# time stepping of $\mathcal{E}_y$, $\mathcal{H}_x$ and $\mathcal{H}_z$
for $t \leq T$ do
    # adapt the grid for $t + \Delta t$ according to ${\mathcal{E}_y}^t$, see the Algorithm 2
    awcm2d_adaptive()
    # ————————————————————————————————
    # update ${\mathcal{H}_x}^{t+\Delta t/2}$, ${\mathcal{H}_z}^{t+\Delta t/2}$ and ${\mathcal{E}_y}^{t+\Delta t}$, see the Algorithm 4
    awcm2d_update()
    # ————————————————————————————————
    # print the data into files for visualization
    awcm2d_print()
    # ————————————————————————————————
    # go to the next time step
    $t = t + \Delta t$
```

We explain here the process of the subroutine awcm2d_adaptive() in detail. Suppose we have a 2D array of $\mathcal{E}_y$ with a mask $Mask0$. We use another mask $pMask0$ to store the information $Mask0$ since $Mask0$ will later be modified by following subroutines. We perform the fast wavelet transform (FWT) of $\mathcal{E}_y$ on the $Mask0$. Note that $Mask0$ is either fully 1 if at the beginning or has been performed a reconstruction check from the last time step, which means the fast wavelet transforms on $Mask0$ are always plausible. Then the 2D array of $\mathcal{E}_y$ will be converted into wavelet domain, i.e., scaling coefficients of the coarsest level j_{min} and wavelet coefficients of levels from j_{min} to $j_{max}-1$. For each wavelet coefficient, we will compare its absolute value with the given tolerance ζ. If the value is less than ζ, we *remove* the corresponding point from the $Mask0$. In this way, $Mask0$ becomes *thinned* or we can say that the information of $Mask0$ is *compressed*. Since some other points which are not in $Mask0$ currently may also become *significant* in the next time step, we add points of adjacent zone to $Mask0$. And then, in order that the FWT of the next time step be possible, we need to perform a reconstruction check to $Mask0$. These two processes are contained in the subroutine Maskext($Mask0$).

After the adaptation of the grid is finished, we go on with the updating fields on adaptive grids. First, we need to update $\mathcal{H}_x$ and $\mathcal{H}_z$ using the spatial derivatives $\dfrac{\partial \mathcal{E}_y}{\partial z}$ and $\dfrac{\partial \mathcal{E}_y}{\partial x}$, respectively. In order to calculate $\dfrac{\partial \mathcal{E}_y}{\partial z}$ and $\dfrac{\partial \mathcal{E}_y}{\partial x}$, we need values of $\mathcal{E}_y$ at neighbor points of points in $Mask0$ which are not in $Mask0$. We store the information of $Mask0$ into $Mask1$. And, we add more points to $Mask1$ needed in calculations of spatial derivatives. For each point in $Mask1$, we add neighbor points needed in calculations of derivatives according to the density level of the points in $Mask1$. We will use a subroutine Level($Mask1$) to calculate *Level*0. *Level*0 is a 2D array that contains information of the density levels of points in $Mask1$. We do not mind the density level of the points which are not in $Mask1$. In order that the inverse wavelet transform be possible, we should again perform a reconstruction check to $Mask1$. This is done by the subroutine gMaskext($Mask1$, *Level*0). After updating of $\mathcal{H}_x$ and $\mathcal{H}_z$, we need to update $\mathcal{E}_y$ using the spatial derivatives $\dfrac{\partial \mathcal{H}_x}{\partial z}$ and $\dfrac{\partial \mathcal{H}_z}{\partial x}$. We should again perform a process of adding neighbor points needed for calculations of spatial derivatives. We store the information of $Mask1$ to $Mask2$. And then we calculate the density level of $Mask2$ and store them into *Level*1. Then perform gMaskext($Mask2$,*Level*1). Now we use IWT($\mathcal{E}_y$, $Mask2$) to reconstruct the values of $\mathcal{E}_y$ in the physical domain.

4.4 Numerical examples

4.4.1 1D Maxwell's equations

We will solve a system of TEM mode equations within an interval $\Omega = [-L/2, L/2]$ in free space, i.e., $\varepsilon_r = 1$ and $\mu_r = 1$. We set the length of the interval $L = 20\mu m$. The initial values for the problem are $\mathcal{E}_y(x, 0) = \exp(-16.0 \times 10^{12} x^2)$ and $\mathcal{H}_z(x, 0) = 0$, for $x \in \Omega$. We will take the minimum and maximum resolution levels as $j_{min} = 5$ and $j_{max} = 10$, respectively. Then the smallest mesh size is $\Delta x = L/2^{j_{max}} = 195.3125nm$. We take the smallest time step $\Delta t = \Delta x/c/1.5$, where $c = 2.99792458 \times 10^8$ is the light speed in the free space. The width of the PML is taken as $L/8$. We take $N = \tilde{N} = 2$, which are the order of ISF and the half of the number of vanishing moments of the lifted wavelet[1], respectively. We take the threshold tolerance $\zeta = 10^{-5}$.

Figure 4.4 shows us the propagation of $\mathcal{E}_y$ field as time evolves. The $\mathcal{E}_y$ field is a Gaussian peak in the center of the interval Ω_0 at the beginning. As time evolves, the field splits into two parts and the two parts propagate in opposite direction and attenuate in the PML region.

We compute the same problem with the adaptive grid method and the full grid method. Figure 4.5 shows the relative error between the two results inside the computational domain which does not include PML region. When the amplitudes of both separated peaks are attenuated by the PML region, then the relative error becomes significant.

[1] The number of vanishing moments of the lifted wavelet is $2\tilde{N}$, see the Theorem 4.5.

Algorithm 2: awcm2d_adaptive for TM$_y$ mode equations

```
# store $Mask0$ into $pMask0$
# $pMask0$: the adaptive grid for $\mathcal{E}_y$ at current time step
$pMask0 = Mask0$
# ————————————————————————————————————————
# fast wavelet transform of $\mathcal{E}_y$ on $Mask0$ with $\zeta$
# $\mathcal{E}_y$ is converted into coefficients of wavelet domain, $Mask0$ is thinned
# see the Algorithm 3
FWT($\mathcal{E}_y$, $Mask0$, $\zeta$)
# ————————————————————————————————————————
# add adjacent zone and perform a reconstruction check to $Mask0$
# see the Algorithm 5
Maskext($Mask0$)
# ————————————————————————————————————————
# add points needed to calculate $\frac{\partial \mathcal{E}_y}{\partial x}$ and $\frac{\partial \mathcal{E}_y}{\partial z}$ on $Mask0$
# 1. determine the density level of each point in $Mask0$
$Level0$ = Level($Mask0$)
# 2. initialize $Mask1$ with $Mask0$
$Mask1 = Mask0$
# 3. update $Mask1$, see the Algorithm 6
gMaskext($Mask1$, $Level$)
# ————————————————————————————————————————
# add points needed to calculate $\frac{\partial \mathcal{H}_x}{\partial z}$ and $\frac{\partial \mathcal{H}_z}{\partial x}$ on $Mask1$
# 1. determine the density level of each point in $Mask1$
$Level1$ = Level($Mask1$)
# 2. initialize $Mask2$ with $Mask1$
$Mask2 = Mask1$
# 3. update $Mask2$, see the Algorithm 6
gMaskext($Mask2$, $Level1$)
# ————————————————————————————————————————
# inverse wavelet transform of the values $\mathcal{E}_y$ in the wavelet domain on $Mask2$
# $\mathcal{E}_y$ is reconstructed from the values in the wavelet domain on $Mask2$
# see the Algorithm 11
IWT($\mathcal{E}_y$, $Mask2$)
```

4.4.2 2D Maxwell's equations

We will solve a system of TM$_y$ mode equations within a square domain $\Omega = [-L/2, L/2] \times [-L/2, L/2]$ in xz plane in free space, i.e., $\varepsilon_r = 1$ and $\mu_r = 1$. We set the domain length $L = 6.0\mu m$. The initial values for the problem are $\mathcal{E}_y(x, z, 0) = \exp(-16.0 \times 10^{12}(x^2 + z^2))$ and $\mathcal{H}_x(x, z, 0) = \mathcal{H}_z(x, z, 0) = 0$ for $x, z \in \Omega$. We will take the minimum and maximum resolution levels as $j_{min} = 3$ and $j_{max} = 9$, respectively. Then the smallest mesh size is

Algorithm 3: FWT

```
Input  : 2D array $A$ of field values, grid mask $Mask\_temp$, $tolerance$
Effects: $A$ will be converted into wavelet domain, the adaptive grid $Mask\_temp$ will
         be thinned by thresholding with $tolerance$.
# ------------------------------------------------------------------------
# calculate the FWT starting from the level $j_{max} - 1$ to the coarsest level $j_{min}$
for $j = j_{max} - 1 : -1 : j_{min}$ do
    # --------------------------------------------------------------------
    # calculation of $d^3_{j,m,n}$
    forall $Q : (x_{j+1,2m+1}, y_{j+1,2n+1}) \in \mathcal{M}^3_j$ do
        if $Q \in Mask\_temp$ then
            calculate $d^3_{j,m,n}$ according to (4.18c)
            if $|d^3_{j,m,n}| < tolerance$ then
                remove the point $Q$ from $Mask\_temp$
    # --------------------------------------------------------------------
    # calculation of $d^1_{j,m,n}$
    forall $Q : (x_{j+1,2m+1}, y_{j,n}) \in \mathcal{M}^1_j$ do
        if $Q \in Mask\_temp$ then
            calculate $d^1_{j,m,n}$ according to (4.18a)
            if $|d^1_{j,m,n}| < tolerance$ then
                remove the point $Q$ from $Mask\_temp$
    # --------------------------------------------------------------------
    # calculation of $d^2_{j,m,n}$
    forall $Q : (x_{j,m}, y_{j+1,2n+1}) \in \mathcal{M}^2_j$ do
        if $Q \in Mask\_temp$ then
            calculate $Mask\_temp$ according to (4.18b)
            if $|d^2_{j,m,n}| < tolerance$ then
                remove the point $Q$ from $Mask\_temp$
    # --------------------------------------------------------------------
    # calculation of $c_{j,m,n}$
    forall $Q : (x_{j,m}, y_{j,n}) \in \mathcal{K}_j$ do
        if $Q \in Mask\_temp$ then
            calculate $c_{j,m,n}$ according to (4.18d)
```

$\Delta = \Delta x = \Delta z = L/2^{j_{max}} = 11.71875nm$. The stability condition of the 2D AWCM scheme is

$$\Delta t \leq \frac{\Delta}{\sqrt{2}c \sum_{l=0}^{l_0-1} |a(l)|}, \tag{4.27}$$

where $a(l)$ is the derivative filters of ISF, see Table 3.1.

The error of the AWCM full grid method is controlled by $O(\Delta t^2)$, since it uses central

Algorithm 4: awcm2d_update for TM$_y$ mode equations

```
# update $\mathcal{H}_x$ and $\mathcal{B}_x$
# 1. interpolate $\mathcal{H}_x$ and $\mathcal{B}_x$ on points in $Mask1$ that is not in $pMask0$
# see the Algorithm 10
interpolate($\mathcal{H}_x$, $pMask0$, $Mask1$)
interpolate($\mathcal{B}_x$, $pMask0$, $Mask1$)
# 2. Calculation of $\frac{\partial \mathcal{E}_y}{\partial z}$ on $Mask1$, similar with the Algorithm 9
$dA_z$ = diffz($\mathcal{E}_y$, $Mask1$, $Level1$, $difffilter$, $dz$)
Update $\mathcal{H}_x$ and $\mathcal{B}_x$ on $Mask1$ using $dA_z$, see for example, (4.26).
# ------------------------------------------------------------------
# update $\mathcal{H}_z$ and $\mathcal{B}_z$
# 1. interpolate $\mathcal{H}_z$ and $\mathcal{B}_z$ on points in $Mask1$ that is not in $pMask0$
interpolate($\mathcal{H}_z$, $pMask0$, $Mask1$)
interpolate($\mathcal{B}_z$, $pMask0$, $Mask1$)
# 2. Calculation of $\frac{\partial \mathcal{E}_y}{\partial x}$ on $Mask1$, see the Algorithm 9
$dA_x$ = diffx($\mathcal{E}_y$, $Mask1$, $Level1$, $difffilter$, $dx$)
Update $\mathcal{H}_z$ and $\mathcal{B}_z$ on $Mask1$ using $dA_x$.
# ------------------------------------------------------------------
# update $\mathcal{E}_y$ and $\mathcal{D}_y$
# 1. interpolate $\mathcal{E}_y$ and $\mathcal{D}_y$ on points in $Mask0$ that is not in $pMask0$
interpolate($\mathcal{E}_y$, $pMask0$, $Mask0$)
interpolate($\mathcal{D}_y$, $pMask0$, $Mask0$)
# 2. Calculations of $\frac{\partial \mathcal{H}_x}{\partial z}$ and $\frac{\partial \mathcal{H}_z}{\partial x}$ on $Mask0$, see the Algorithm 9
$dA_z$ = diffz($\mathcal{H}_x$, $Mask0$, $Level0$, $difffilter$, $dz$)
$dA_x$ = diffx($\mathcal{H}_z$, $Mask0$, $Level0$, $difffilter$, $dx$)
Update $\mathcal{E}_y$ and $\mathcal{D}_y$ on $Mask0$ using $dA_z$ and $dA_x$.
```

Algorithm 5: Maskext

```
Input  : a grid mask $Mask\_temp$
Effects: the grid mask $Mask\_temp$ will be modified
# add adjacent zone to $Mask\_temp$, see the Subsection 4.3.2
Maskext_adj($Mask\_temp$)
# ------------------------------------------------------------------
# perform the reconstruction check to $Mask\_temp$, see the Algorithm 8
Maskext_rec($Mask\_temp$)
```

difference of second order for discretization of the time derivatives. Our choice of the threshold tolerance should be larger than the discretization error of AWCM. We take $\Delta t = \Delta/c/1.6$, which is a little bit smaller than the upper bound of the CFL stability condition. And we take a smaller threshold, $\zeta = 5.0 \times 10^{-4}$. The PML width is taken as $L/4$. Let N be the order of ISF and $\tilde{N}$ be the half of the number of vanishing moments of the lifted

Algorithm 6: gMaskext

```
Input  : a grid mask $Mask\_temp$, density level $Lev$
Effects: the grid mask $Mask\_temp$ will be modified
# add points, needed to calculate derivatives, to $Mask\_temp$, see the Algorithm 7
Maskext_diff($Mask\_temp$, $Lev$)
# ------------------------------------------------------------
# perform the reconstruction check to $Mask\_temp$, see the Algorithm 8
Maskext_rec($Mask\_temp$)
```

Algorithm 7: Maskext_diff

```
Input  : a grid mask $Mask\_temp$, density level $Lev$
Effects: the grid mask $Mask\_temp$ will be modified
# ------------------------------------------------------------
# initialize $Mask\_temp1$ with $Mask\_temp$
$Mask\_temp1 = Mask\_temp$
# ------------------------------------------------------------
# add points, needed for calculation of derivatives, to $Mask\_temp$
forall $Q = (x_{j_{max},m}, y_{j_{max},n}) \in \mathcal{K}_{j_{max}}$ do
    if $Q \in Mask\_temp$ then
        # read the density level of $Q$ from the input $Lev$
        $j_0 = Lev[n][m]$
        include the closest $2N$ neighboring points of $Q$ in the level $j_0$ into $Mask\_temp$
```

wavelet. We take $N = \tilde{N} = 4$.

The centered Gaussian peak will spread away from the center of the domain Ω as time evolves. Unlike the 1D case, since the energy is spreading along each direction in 2D, the amplitude is decreasing, while in 1D case, the amplitude will stay as $1/2$ during propagation. We can see the compression of the grid points from Figure 4.6.

Figure 4.8 shows the relative error of $\mathcal{E}_y$ field values between adaptive and full grid methods only inside the computational domain as time evolves. The amplitude of a 2D Gaussian decays as it propagates towards the boundary. When the amplitude approaches the threshold tolerance ζ, the relative error of $\mathcal{E}_y$ field between adaptive and full grid methods increases.

We record computation time for every ten time steps. The computation of time stepping is fast when there is a relatively less number of adaptive grid points. The time evolution profile of the computation time is very similar to that of the percentage of adaptive grid points in adaptive grid, see Figure 4.9. We know from numerical experiments that the computation time of full grid method for every ten time stepping is about 18 seconds. In adaptive grid method, the computation time of ten time steps becomes greater than 18 seconds when the percentage of adaptive grid points is more than about 2.5%. In Figure 4.9, we see that the percentage is lower than 2.5% for most of the time steps. Therefore, adaptive grid method is faster than full grid method for this example. It took 1900 seconds

Algorithm 8: Maskext_rec

```
Input  : a grid mask $Mask\_temp$
Effects: the grid mask $Mask\_temp$ will be modified
# ----------------------------------------------------------------
# $j$: resolution level
for $j = j_{max} - 1 : -1 : j_{min}$ do
    # ------------------------------------------------------------
    # add points around $d^3_{j,m,n}$
    forall $Q = (x_{j+1,2m+1}, y_{j+1,2n+1}) \in \mathcal{M}^3_j$ do
        if $Q \in Mask\_temp$ then
            include neighbor points of $Q$ into $Mask\_temp$ according to Figure 4.1(c)
    # ------------------------------------------------------------
    # add points around $d^1_{j,m,n}$
    forall $Q = (x_{j+1,2m+1}, y_{j,n}) \in \mathcal{M}^1_j$ do
        if $Q \in Mask\_temp$ then
            include neighbor points of $Q$ into $Mask\_temp$ according to Figure 4.1(a)
    # ------------------------------------------------------------
    # add points around $d^2_{j,m,n}$
    forall $Q = (x_{j,m}, y_{j+1,2n+1}) \in \mathcal{M}^2_j$ do
        if $Q \in Mask\_temp$ then
            include neighbor points of $Q$ into $Mask\_temp$ according to Figure 4.1(b)
```

Algorithm 9: diffx

```
# $Level$: $x$ level of each point in $Mask\_temp$, $difffilter$: see table 3.1, $h$: the
smallest mesh size at the highest resolution level
Input  : a 2D array of field $A$, a grid mask $Mask\_temp$, $Level$, $difffilter$, $h$
Return: a 2D array of $\frac{\partial A}{\partial x}$ on $Mask\_temp$
# initialize a 2D array $dA$ for the storage of $\frac{\partial A}{\partial x}$
$dA = 0$
forall $Q = (x_{j_{max},m}, y_{j_{max},n}) \in \mathcal{K}_{j_{max}}$ do
    if $Q \in Mask\_temp$ then
        # read the density level of $Q$ from $Level$
        $j(Q) = Level[n][m]$
        calculate $dA$ at point $Q$ using $difffilter$ and values of $A$ at neighbor points
        in the level $j(Q)$. See the equation (4.24).
```

to run 1200 time steps with adaptive grid method, while 2160 seconds were used with full grid method. However, this example is an extreme case. In general, we cannot expect every example to have such nice compression, i.e., greater than 97.5%. We say the AWCM is still

```
Algorithm 10: interpolate
  # interpolate field values of A at points in $Mask\_temp2$ that is not included in $Mask\_temp1$
  Input : a 2D array of field $A$, a grid mask $Mask\_temp1$, a grid mask $Mask\_temp2$
  Effects: $A$ is modified by the interpolation
  # initialize an empty grid mask $Mask\_temp0$
  $Mask\_temp0 \leftarrow 0$
  # include points in $Mask\_temp2$ but not in $Mask\_temp1$
  forall $Q = (x_{j_{max},m}, y_{j_{max},n}) \in \mathcal{K}_{j_{max}}$ do
      if $Q \in Mask\_temp2$ but $\notin Mask\_temp1$ then
          include $Q$ into $Mask\_temp0$
  # perform the reconstruction check to $Mask\_temp0$
  Mask_rec($Mask\_temp0$)
  for $j = j_{min} : j_{max} - 1$ do
      # interpolate $A$ at $(x_{j+1,2m+1}, y_{j,n})$
      forall $Q = (x_{j+1,2m+1}, y_{j,n}) \in \mathcal{M}_j^1$ do
          if $Q \in Mask\_temp0$ but $\notin Mask\_temp1$ then
              calculate $c_{j+1,2m+1,2n}$ according to (4.19b) with $d_{j,m,n}^1 = 0$
      # interpolate $A$ at $(x_{j,m}, y_{j+1,2n+1})$
      forall $Q = (x_{j,m}, y_{j+1,2n+1}) \in \mathcal{M}_j^2$ do
          if $Q \in Mask\_temp0$ but $\notin Mask\_temp1$ then
              calculate $c_{j+1,2m,2n+1}$ according to (4.19c) with $d_{j,m,n}^2 = 0$
      # interpolate $A$ at $(x_{j+1,2m}, y_{j+1,2n})$
      forall $Q = (x_{j+1,2m}, y_{j+1,2n}) \in \mathcal{M}_j^3$ do
          if $Q \in Mask\_temp0$ but $\notin Mask\_temp1$ then
              calculate $c_{j+1,2m,2n}$ according to (4.19d) with $d_{j,m,n}^3 = 0$
```

computationally efficient for examples of which compression is about 90%, because of the huge reduction of the storage of data, although the total computation time is longer than that of full grid method.

Algorithm 11: IWT

```
Input  : a 2D array $A$, a grid mask $Mask\_temp$
Effects: values of $A$ at points in $Mask\_temp$ will be converted into physical domain
for $j = j_{min} : j_{max} - 1$ do
    # calculate $c_{j+1,2m,2n}$
    forall $Q = (x_{j+1,2m}, y_{j+1,2n}) \in \mathcal{K}_j$ do
        if $Q \in Mask\_temp$ then
            calculate $c_{j+1,2m,2n}$ according to (4.19a)
        else
            $A$ at point $(x_{j+1,2m}, y_{j+1,2n}) = 0$
    # calculate $c_{j+1,2m+1,2n}$
    forall $Q = (x_{j+1,2m+1}, y_{j+1,2n}) \in \mathcal{M}_j^1$ do
        if $Q \in Mask\_temp$ then
            calculate $c_{j+1,2m+1,2n}$ according to (4.19b)
        else
            $A$ at point $(x_{j+1,2m+1}, y_{j+1,2n}) = 0$
    # calculate $c_{j+1,2m,2n+1}$
    forall $Q : (x_{j+1,2m}, y_{j+1,2n+1}) \in \mathcal{M}_j^2$ do
        if $Q \in Mask\_temp$ then
            calculate $c_{j+1,2m,2n+1}$ according to (4.19c)
        else
            $A$ at point $(x_{j+1,2m}, y_{j+1,2n+1}) = 0$
    # calculate $c_{j+1,2m+1,2n+1}$
    forall $Q : (x_{j+1,2m+1}, y_{j+1,2n+1}) \in \mathcal{M}_j^3$ do
        if $Q \in Mask\_temp$ then
            calculate $c_{j+1,2m+1,2n+1}$ according to (4.19d)
        else
            $A$ at point $(x_{j+1,2m+1}, y_{j+1,2n+1}) = 0$
```

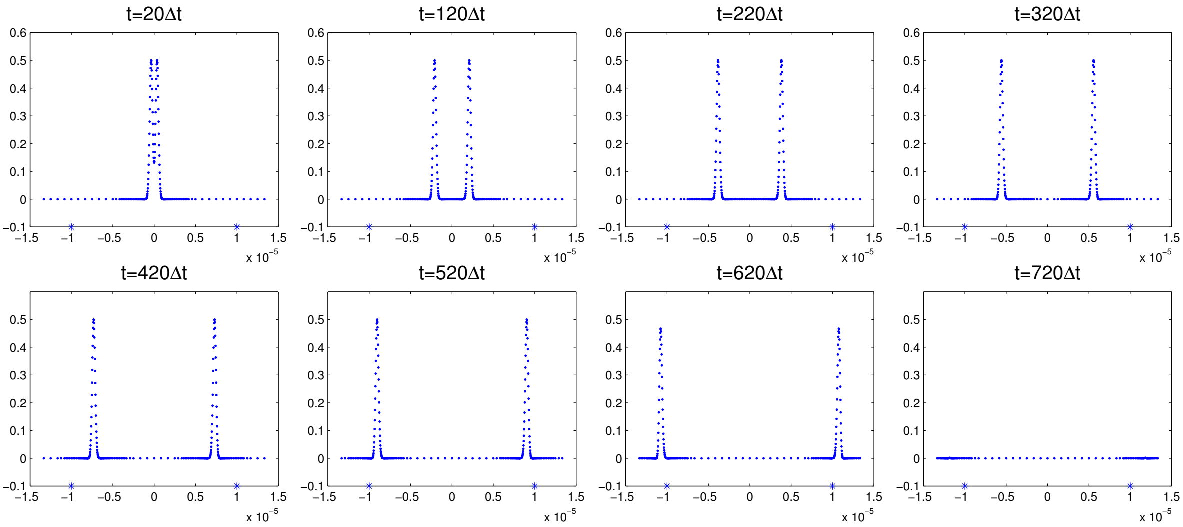


Figure 4.4: Visualization of the propagation of a 1D Gaussian peak along the x axis. Note that the main computational domain is the interval $[-1.0\times10^{-5}, 1.0\times10^{-5}]$, and the intervals $[-1.5\times10^{-5}, -1.0\times10^{-5}]$ and $[1.0\times10^{-5}, 1.5\times10^{-5}]$ are both PML regions.

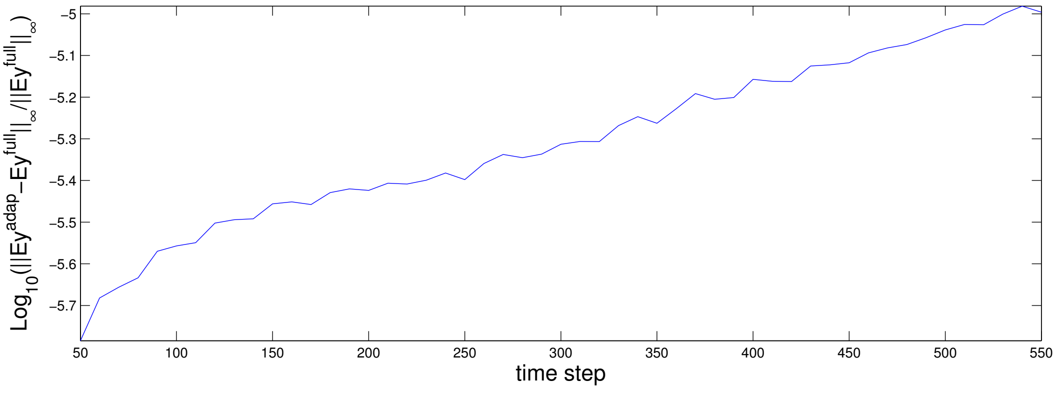


Figure 4.5: Relative error of $\mathcal{E}_y$ field between adaptive grid and full grid method: $N = \tilde{N} = 2$, $\zeta = 10^{-5}$.

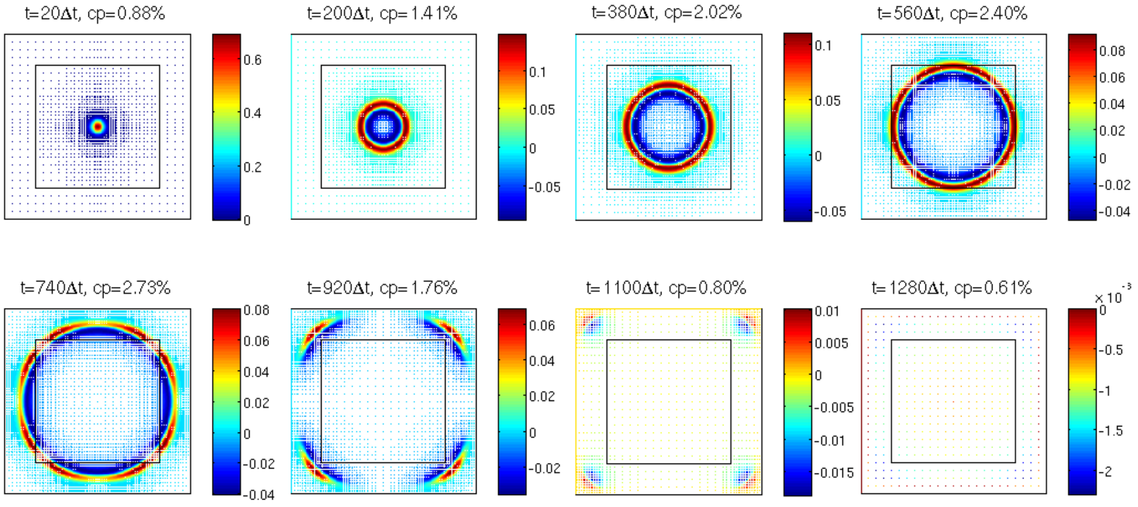


Figure 4.6: Visualization of the propagation of a 2D Gaussian peak $\mathcal{E}_y$ in the cross-section plane xz: $N = \tilde{N} = 4$, $\zeta = 5.0 \times 10^{-4}$, cp: percentage of adaptive grid points. For animation movie, see the Youtube channel: http://www.youtube.com/user/HaojunLi?feature=mhee#p/u/1/2Yzpjf7Xnp4.

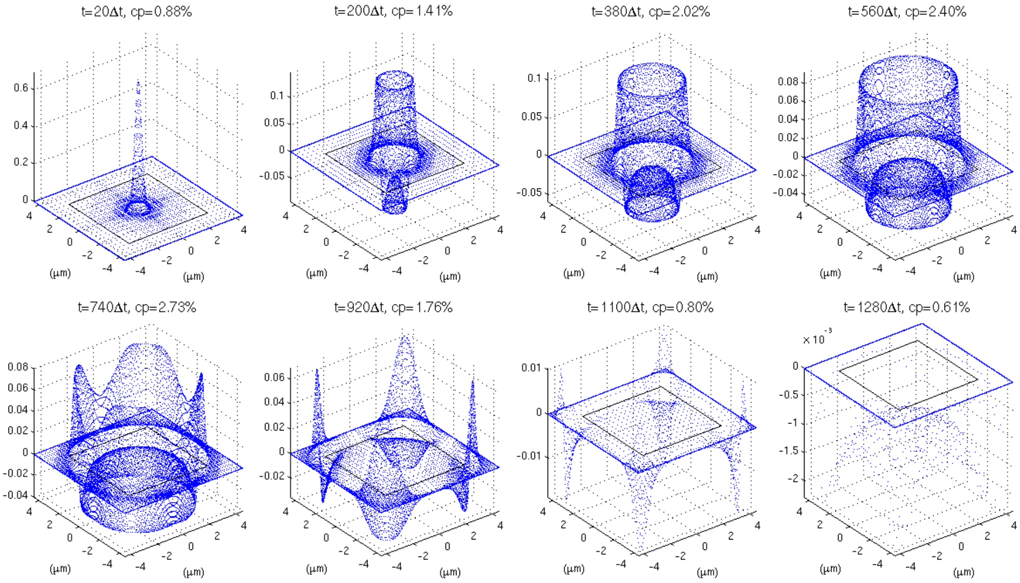


Figure 4.7: Visualization of the propagation of a 2D Gaussian peak $\mathcal{E}_y$ in space: $N = \tilde{N} = 4$, $\zeta = 5.0 \times 10^{-4}$, cp: percentage of adaptive grid points.

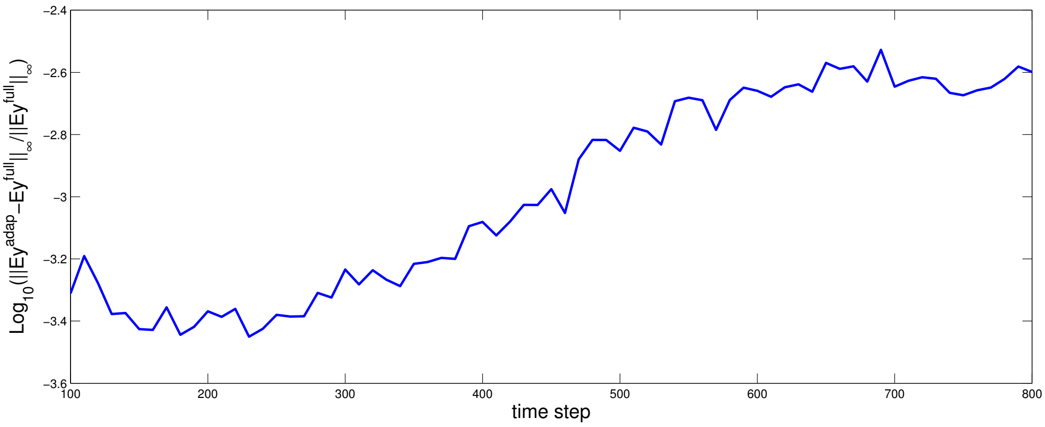


Figure 4.8: Relative Error of $\mathcal{E}_y$ between the adaptive and full grid method : $N = \tilde{N} = 4$, $\zeta = 5.0 \times 10^{-4}$.

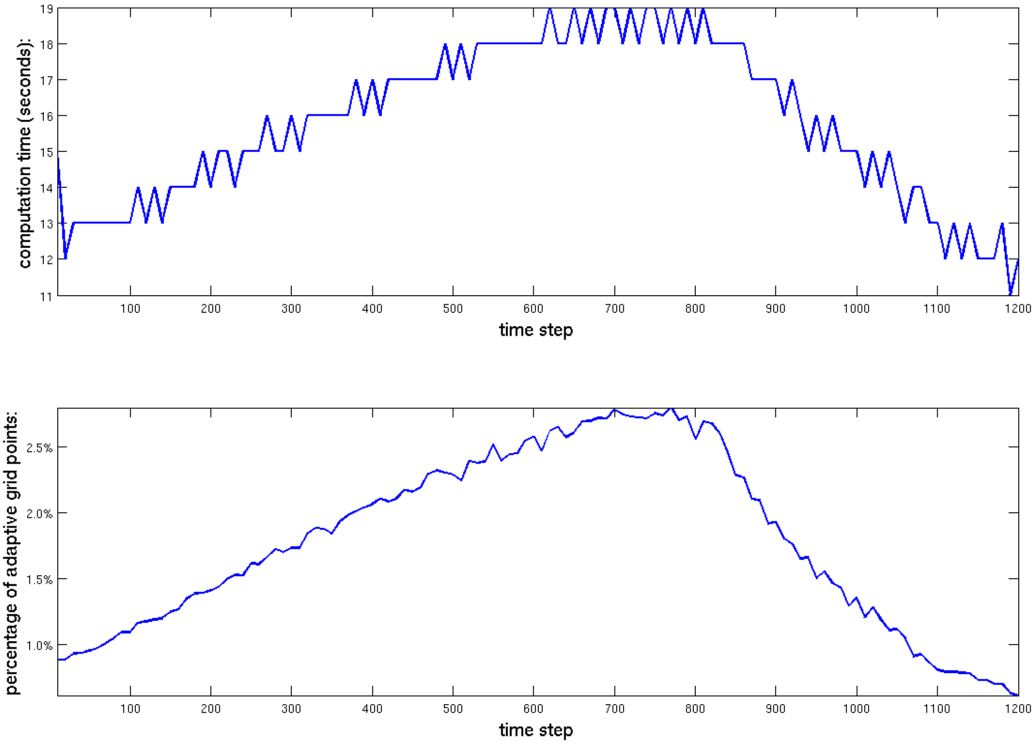


Figure 4.9: Time evolution profiles of computation time and that of percentage of adaptive grid points. The computation time was recorded for every ten time steps.

Chapter 5

Simulation of the micro-ring resonator

In this chapter, we will investigate the applicability of adaptive wavelet collocation method (AWCM) to simulation of the micro-ring resonator. The mathematical modeling of this optical device has been described in Chapter 1. The purpose of the simulation is to find out the frequency responses of ring resonators. This is done by launching a guided signal through the straight waveguide below which contains a bundle of frequencies to check how it interacts with the ring cavity (See Figure 1.1). We set several checkpoints in the waveguides to store signal values during time stepping. We apply discrete Fourier transform (DFT) to the stored values of the signal at those checkpoints. In order to obtain a fair frequency response profile, we need to run the simulation code for a certain number of time steps to store values. Then we can find out resonance frequencies of the ring-resonator, and compare the results with that of obtained with other available methods like finite difference time domain (FDTD) method (Chapter 2), interpolating scaling functions method (ISFM) (Chapter 3) and coupling mode theory (CMT) method [17]. Subsequently we also characterize the resonances by various measures like the free spectral range (FSR), the quality factor (Q factor) etc.

5.1 Source excitation

We will launch two types of sources. One is a guided mode of slab waveguide for a specific frequency, the other one is a Gaussian pulse modulating a frequency carrier. The Gaussian pulse modulating a frequency carrier is used to check out the resonance frequencies of the device and the mono-frequency guided mode of the slab waveguide is used to confirm the on-off resonance behavior of the resonator for the resonant frequencies.

Before we discuss the type of source called Gaussian pulse modulating a frequency carrier, we need to understand first how to derive transverse profiles of a guided mode of a dielectric straight waveguide at a single frequency. The theory of guided mode of a dielectric straight waveguide is a well established topic. See Appendix for the detail of derivation of the field profiles of the guided mode of dielectric waveguide.

5.1.1 Gaussian pulse modulating a frequency carrier

In order to check the frequency response of the resonator, we need to launch a type of source that contains a range of frequencies about a center frequency ω_c. We can see from the Appendix that the longitudinal propagation constant β depends on the choice of the angular frequency ω. Here we will derive the accurate Gaussian pulse, see for example, X. Ji [18]. We assume the time dependence of the source to be

$$f(t) = \exp(\imath\omega_c(t - t_0) - (t - t_0)^2/t_{decay}^2), \quad t > 0, \tag{5.1}$$

where t_0 is the centered time of the Gaussian pulse and t_{decay} (> 0) is a scaling factor of the pulse. The Fourier transform of f is

$$\hat{f}(\omega) = \frac{t_{decay}}{\sqrt{2}} \exp(-(\omega - \omega_c)^2 t_{decay}^2/4 - \imath t_0\omega). \tag{5.2}$$

Assume $\mathbf{F} \in \{\mathbf{E_y}, \mathbf{H_x}, \mathbf{H_z}\}$ to be the corresponding phasor profile of the mono-frequency mode $\exp(\imath\omega t)$ as calculated in Appendix, we obtain by integration

$$\mathcal{F}(x, z, t) = \frac{1}{\sqrt{2\pi}} \int \mathbf{F}(x, z, \omega)\hat{f}(\omega) \exp(\imath\omega t) d\omega, \tag{5.3}$$

where $\mathcal{F} \in \{\mathcal{E}_y, \mathcal{H}_x, \mathcal{H}_z\}$ is a field in the time domain.

Then, $\mathcal{F}$ is an accurate Gaussian pulse with frequencies centered at ω_c, which satisfies the TM$_y$ mode equations (2.10). In our simulation, we approximate the phasor profile with different frequencies with that at the center frequency ω_c for simplicity. Therefore, the time dependent source function of our simulation is

$$\mathcal{F}(x, z, t) = \mathbf{F}(x, z, \omega_c) f(t).$$

5.1.2 TF/SF formulation with AWCM

The total field and scattered field (TF/SF) formulation allows us modulating incident sources of long-duration such as continuous mono-frequency waves. If we use a hard source method to launch such type of long-duration sources, then there will occur retro-reflections at the sourcing position when the waves scattered by materials propagate back to the sourcing position. And these retro-reflections will introduce contaminations into the computational domain. However, the TF/SF formulation of incident sources is an analytical way of launching source. In particular, if we use hard sources in AWCM, such retro-reflections will cause bad compressions of grid points. Therefore, we choose the TF/SF method to launch sources, although it is more complicated than the hard source method in implementation.

The idea of the total field and scattered field (TF/SF) formulation of incident sources ([38],[35]) is based on the linearity of Maxwell's equations. Assume that we know an accurate incident wave beforehand. We decompose the total field as a combination of the incident field and the scattered field.

$$\begin{aligned}
\mathcal{E}_{y,total} &= \mathcal{E}_{y,inc} + \mathcal{E}_{y,scat}, \\
\mathcal{H}_{x,total} &= \mathcal{H}_{x,inc} + \mathcal{H}_{x,scat}, \\
\mathcal{H}_{z,total} &= \mathcal{H}_{z,inc} + \mathcal{H}_{z,scat}.
\end{aligned}$$

In the total field region the total field values are stored and in the scattered field region the scattered field values are stored. Then when we approximate the derivatives with AWCM at the numerical grid points near the TF/SF interface, both of total field and scattered field values are used. We must correct these numerical derivatives obtained by mixed types of field values. For example, we will look at the equation for updating of $\mathcal{H}_z$ component,

$$\frac{\partial \mathcal{H}_z}{\partial t} = -\frac{1}{\mu}\frac{\partial \mathcal{E}_y}{\partial x}. \tag{5.4}$$

We artificially include the finest grid points near the interface between TF/SF into the adaptive grid for convenience. Therefore, the density level of the grid points near the interface is always the maximum, j_{max}. Suppose $\Delta x = L/2^{j_{max}}$. For a grid point $Q = (x_{j_{max},m'}, y_{j_{max},n'})$, we use $\mathcal{A}|_Q^k$ to denote the value of $\mathcal{A}$ at the point Q in the time step $k\Delta t$. Then the AWCM discretization of the equation (5.4) is

$$\mathcal{H}_z|_Q^{k+1/2} = \mathcal{H}_z|_Q^{k-1/2} - \frac{\Delta t}{\mu_0 \Delta x}\sum_m \mathcal{E}_y|^k_{(x_{j_{max},m},y_{j_{max},n'})}(DD_N)'(m'-m). \tag{5.5}$$

We consider the case $N = 2$, then, (5.5) becomes

$$\begin{aligned}\mathcal{H}_z|_Q^{k+1/2} =& \mathcal{H}_z|_Q^{k-1/2} - \frac{\Delta t}{\mu_0\Delta x}\bigg(\frac{1}{12}\mathcal{E}_y|^k_{(x_{j_{max},m'-2},y_{j_{max},n'})} - \frac{2}{3}\mathcal{E}_y|^k_{(x_{j_{max},m'-1},y_{j_{max},n'})} \\ &+ \frac{2}{3}\mathcal{E}_y|^k_{(x_{j_{max},m'+1},y_{j_{max},n'})} - \frac{1}{12}\mathcal{E}_y|^k_{(x_{j_{max},m'+2},y_{j_{max},n'})}\bigg).\end{aligned} \tag{5.6}$$

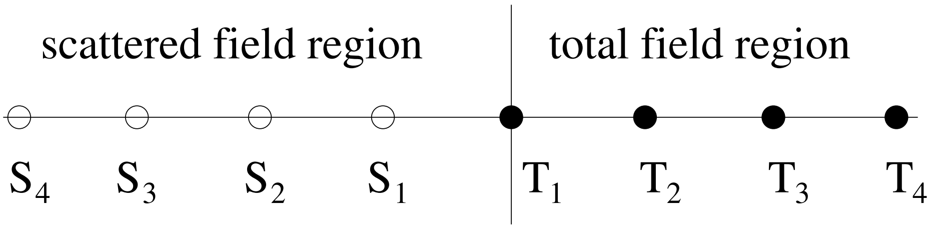


Figure 5.1: Description of numerical grid points for total field and scattered field.

Let us look at Figure 5.1. When Q is in the scattered field region, for example, $Q = S_2$, the equation (5.6) becomes

$$\mathcal{H}_z|_{S_2}^{k+1/2} = \mathcal{H}_z|_{S_2}^{k-1/2} - \frac{\Delta t}{\mu_0\Delta x}\bigg(\frac{1}{12}\mathcal{E}_y|^k_{S_4} - \frac{2}{3}\mathcal{E}_y|^k_{S_3} + \frac{2}{3}\mathcal{E}_y|^k_{S_1} - \frac{1}{12}\mathcal{E}_y|^k_{T_1}\bigg). \tag{5.7}$$

We implement the TF/SF formulation by modifying the normal AWCM code, i.e., in which all the fields are total fields, during each time-stepping. For example, before modifying, the $\mathcal{H}_z$ component is *falsely updated* according to the equation (5.7) as:

$$\begin{aligned}\{\mathcal{H}_{z,scat}|_{S_2}^{k+1/2}\} =& \mathcal{H}_{z,scat}|_{S_2}^{k-1/2} - \frac{\Delta t}{\mu_0\Delta x}\bigg(\frac{1}{12}\mathcal{E}_{y,scat}|^k_{S_4} - \frac{2}{3}\mathcal{E}_{y,scat}|^k_{S_3} \\ &+ \frac{2}{3}\mathcal{E}_{y,scat}|^k_{S_1} - \frac{1}{12}\mathcal{E}_{y,total}|^k_{T_1}\bigg),\end{aligned} \tag{5.8}$$

where $\left\{\mathcal{H}_{z,scat}|_{S_2}^{k+1/2}\right\}$ is not a correct value, since the stored value of $\mathcal{E}_y$ at the point T_1 is a total field, we must correct this term with a scattered field. Since the scattered field is the difference between the total field and the incident field, we can get the right value by

$$\mathcal{H}_{z,scat}|_{S_2}^{k+1/2} = \left\{\mathcal{H}_{z,scat}|_{S_2}^{k+1/2}\right\} - \frac{1}{12}\frac{\Delta t}{\mu_0 \Delta x}\mathcal{E}_{y,inc}|_{T_1}^{k}. \tag{5.9}$$

In the same way, we can correct all of the corresponding field values at S_1, T_1 and T_2.

Remark 5.1. *In time-stepping of the* TM_y *equations with AWCM, we need to interpolate values on some adaptive grid points which were not present in the adaptive grid of the previous time step. The interpolation is based on the Lagrangian interpolation with the neighboring points. If those points are on the interfaces close to the TF/SF interface, then, the interpolations are not correct since they involve both total field points and scattered field points. It is not impossible that we correct these values as TF/SF method above.*

However, these may enhance the complicatedness of the code. A remedy for this problem is to keep sufficiently "thick" layers close to the TF/SF interface the finest level so that the interpolations do not happen at the boundary of the different type of the fields. Another simpler way to avoid this is to turn off adaptivity of the grid, i.e., full grid calculation without FWT and IWTs, until the incident source stops launching.

5.2 Numerical simulations of the micro-ring resonator with AWCM

In this section, we will perform the numerical simulation of the micro-ring resonator (Figure 5.2) with AWCM and discuss the simulation results. We will compare AWCM with FDTD, interpolating scaling functions method (ISFM) and coupled mode theory (CMT) [17] on these simulation results.

We simulate the setting as in [16, 35]. We launch a 20-fs full width at half maximum (FWHM) Gaussian pulse modulating a 200-THz carrier (center frequency ω_c) at the left port A of the straight waveguide WG1 (Figure 5.2) using the TF/SF formulation. Then the incident Gaussian pulse propagates along WG1. When it reaches the region close to the ring, it interacts with the ring and some parts of the Gaussian pulse switch into the ring and the rest of the pulse continues to propagate and exits from the right port B of WG1. The signal which switched into the ring cavity continues to circumnavigate inside the ring and some parts of the signal will interact with WG2 and exit from the left port C of WG2.

In the simulation, the outer radius R of the ring is 2.5µm and the width wr of the ring is 0.3µm. The width ws of the straight waveguide is 0.3µm and the gap distance g is 0.232µm. The refractive indices of the straight waveguide and the ring cavity are both 3.2, i.e., $n_s = n_c = 3.2$. We take a squared computational domain with the length $L = 8$µm and the width of the PML layer $L/4$. See Figure 5.3 for the geometry of the computational domain, PML layer and TF/SF interface.

We take the maximum and minimum resolution level, $j_{max} = 9$, $j_{min} = 3$, respectively. Then the smallest mesh size, $\Delta = \Delta x = \Delta z$, is $15.625 nm$. And we take $N = \tilde{N} = 2$.

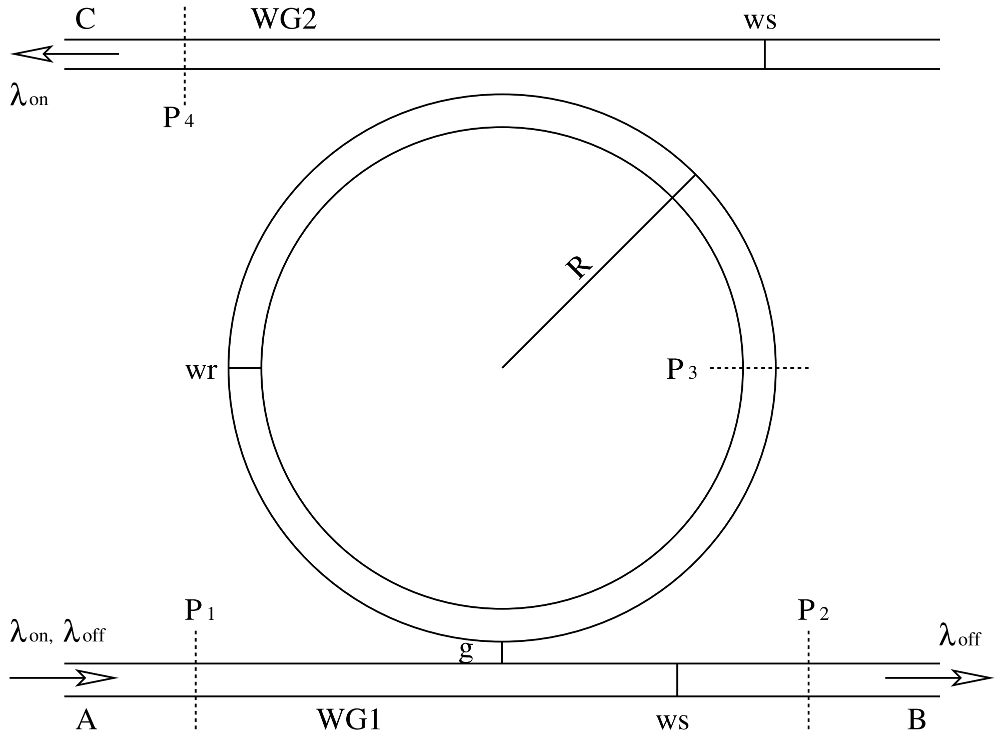


Figure 5.2: A geometric diagram of a micro-ring resonator, which is composed of a circular ring cavity and two lateral straight waveguides. On-resonance and off-resonance signal excited from port A are guided with different directions by the ring resonator. Source: [35].

The smallest time step size is taken according to the CFL stability condition[1], i.e., $\Delta t \leq \frac{2\sqrt{2}\Delta}{3c}$. Since AWCM uses a central difference scheme of second order to discretize the time derivatives of the TM$_y$ equations, the error of the full grid solutions to our problem is limited by $O(\Delta t^2)$. We choose a smaller time step size, i.e., $\Delta t = \frac{\Delta}{1.6c}$, in order to take a relatively smaller threshold tolerance, $\zeta = 5.0 \times 10^{-4}$.

The choice of the threshold tolerance is the most crucial in the whole simulation. If we take a threshold which is too small, then the adaptivity efficiency is very bad, i.e., almost all the points of the finest level are in the adaptive grid. And if the threshold is large, then, after the incident source exits from the right port of WG2, all the grid points are compressed, thus, only the points of the coarsest level will remain in the adaptive grid. Therefore, we cannot obtain detailed information for the signal circumventing inside the ring.

We run the simulation 2^{18} time steps. The reason we select the number of simulation time steps as a power of 2 is because we want to perform FFT with stored field values.

The snapshots of several time steps of the $\mathcal{E}_y$ field component are plotted in Figures 5.4 and 5.5. We see that the numerical dense grid points are *following* the intensive signal in the ring. Note that the points of the cross-sections of the four ports are always kept as the finest resolution level in order that we store the field information on these cross-sections.

[1] The stability condition for the AWCM is essentially the same as that of ISFM, see (3.38).

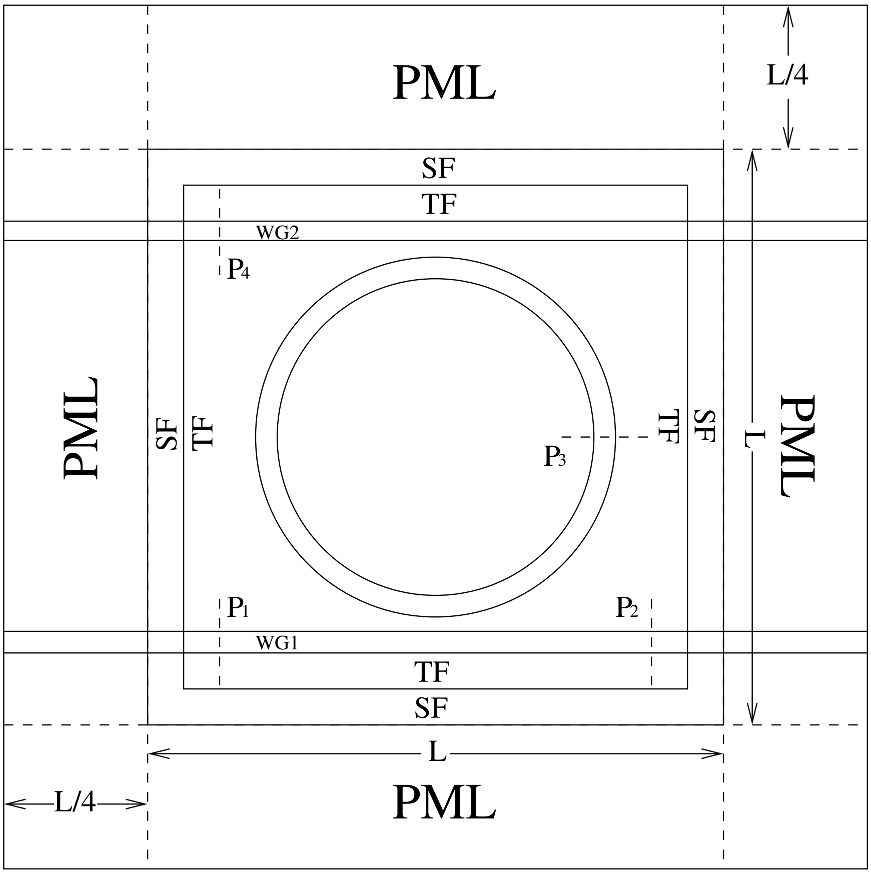


Figure 5.3: A geometric diagram of the computational domain, PML region and TF/SF interface for the simulation of micro-ring resonator.

The percentage *cp* of the adaptive grid points are given in the title of each snapshot with the time step. Note that we only plotted fields in computational domain. For information on fields in PML see the animation online. The link is provided in the caption of Figure 5.5.

5.2.1 Spectrum response

The signal which switched into the ring continues to circumnavigate inside the ring until all its energy extinguishes. We store the information of the signal at the cross-sections P1, P2, P3 and P4 during time stepping. We use the stored time domain field components to calculate the average power densities flux through the cross-sections. The average power density[2] is defined as

$$\mathcal{P}_{av} = \frac{1}{2}\Re\big[\mathbf{E}\times\mathbf{H}^*\big]\Big|_{\mathrm{P}_e}, \qquad e \in \{1,2,3,4\}, \tag{5.10}$$

where $\mathbf{E}$ and $\mathbf{H}$ are the phasors in the frequency domain, and P_e ($e = 1,2,3,4$) are the cross-section ports for recording data.

[2] See more detail about Poynting vector and power in [3].

The length of the cross-section P_e $(e = 1, 2, 3, 4)$ should be taken big enough so that the average power density flux calculated on the corss-section represents almost the full modal power density. In our simulation we take the length of the cross-section four times the width of the ring. We integrate the power density flux on the cross-section P_3 and normalize it by that on the cross-section P_1 to get the *normalized dropped power* as a function of frequency, see Figures 5.6 and 5.7. The sharp peaks in these figures are resonant frequencies. We also calculate the *normalized transmitted power* which is defined as the ratio between the power density flux on the cross-section P_2 and that on the cross-section P_1, see Figure 5.8. In these figures, we rescaled the normalized power by its maximum in order to have highly contrasted view of the resonant frequencies.

5.2.2 Coupling efficiency

The coupling coefficient κ is defined as the ratio between the power switched into the ring and that of the incident power. It is the percentage of power coupled into the ring from WG1. Since the power switched into the ring continues to circumvent inside the ring, we should calculate the switched power with the stored fields at the cross-section P3 only for the first circulation.

We can see from Figure 5.9 that the coupling efficiency decreases as the frequency increases and the gap size distance increases.

5.3 Comparison with other methods

In this section, we compare the resonant data obtained with different methods, such as FDTD, ISFM, CMT. We perform FFT on the 2^{19} equally spaced time samples obtained by FDTD and ISFM, see Figure 5.10, 5.11.

Calculating the resonant frequencies with single FFT is time consuming. Since the frequency resolution is reciprocal to the product of the total number of time-steps and the time-step size, thus, in order to get a desired accuracy of the resonant frequencies, the simulation time should be sufficiently long. There are several methods, such as Prony's method [29], Pade approximation [11], which can obtain nice resonant frequencies with relatively shorter window of time domain fields by extracting FFT peaks. We don't want to discuss these in detail here. There is a free software, called Harminv [26], which can obtain much better accuracy than straightforwardly extracting FFT peaks [19]. We use it to calculate the resonant data from the stored window of time domain field values obtained with several different methods and compare them. The stored time domain samples are combinations of decaying modes in the form,

$$\exp((\alpha + 2\pi\imath f)t), \tag{5.11}$$

where α is the decay constant and f is the frequency of the mode. We define the quality factor (Q factor) [3] of the mode (5.11) to be

$$Q := \frac{\pi|f|}{|\alpha|}. \tag{5.12}$$

The resonant frequencies are those with high Q factors. We perform Harminv [26] on the 2^{15} samples of the time domain electric field obtained by AWCM, FDTD, ISFM to get frequencies with Q factors.

Let m be the longitudinal mode number inside the ring, i.e., the number of wavelengths in the ring. Table 5.1 shows resonant data of the same micro-ring resonator calculated by Harminv [26], except the coupled mode theory method (CMT), which is provided by K. R. Hiremath, [17][4]. We see from the table that the resonant frequencies of the same mode obtained by different methods are differed by each other only within 0.3% range.

Table 5.1: Comparison of the resonance data for the 5.0μm-Diameter ring-resonator obtained with four different methods. m: longitudinal mode number in the ring; f_m: resonant frequency; λ_m: resonant wavelength; Q: Q factor; FSR: free spectral range.

(a) AWCM

m	f_m(THz)	λ_m (nm)	Q	FSR (nm)
25	186.15	1610.51	3206.00	50.99
26	192.23	1559.52	2068.65	47.62
27	198.29	1511.90	23185.0	44.69
28	204.33	1467.21	3979.14	42.32
29	210.40	1424.89	5130.40	

(b) FDTD

m	f_m(THz)	λ_m (nm)	Q	FSR (nm)
25	186.04	1611.48	3509.95	51.09
26	192.13	1560.39	4364.47	47.09
27	198.11	1513.30	5918.55	44.87
28	204.16	1468.43	7817.19	41.91
29	210.16	1426.52	11328.7	

(c) ISFM

m	f_m(THz)	λ_m (nm)	Q	FSR (nm)
25	186.19	1610.11	3499.43	51.23
26	192.31	1558.88	3692.44	47.04
27	198.30	1511.84	6368.16	44.94
28	204.37	1466.90	11301.4	41.97
29	210.39	1424.93	13771.4	

(d) CMT [17]

m	f_m(THz)	λ_m (nm)	Q	FSR (nm)
25	185.85	1613.1	4473	53.5
26	192.23	1559.6	5776	44.7
27	197.90	1514.9	7560	43.4
28	203.73	1471.5	10482	43.5
29	209.94	1428.0	14252	

[3] Sometimes Q is defined as ratio between $\pi|f|$ and the full width at half maximum of the corresponding frequency [17].

[4] We calculate f_m's and (FSR)s in Table 5.1(d) from λ_m in [17]. And Q's are provided by K. R. Hiremath.

5.3.1 Steady state resonances

We can find the resonant frequencies from the spectrum response (Table 5.1) of the ring resonator, or those peaks in Figures 5.6 and 5.7 are resonant frequencies.

If we continuously launch guided modes of the straight waveguide at these resonant frequencies, we will observe intensive resonant power energy inside the ring. We should run a sufficient number of time steps to simulate the steady state of on-resonances. The number of time steps needed to reach the steady state of an on-resonance depends on the frequency and the gap distance between the straight waveguides and the ring cavity. We have seen that the coupling efficiency is higher with smaller frequency and smaller gap distances, see Figure 5.9. If the coupling efficiency is high, for example, above 50%, then we can reach the steady-state of the on-resonance with only several round trips of the wave inside the ring. But the intensity of the resonance in the steady state is higher for a frequency of the signal and a gap distance of the ring with low coupling efficiency. In this case, we need to run the simulation for hundreds or thousands of round trips. When m increases, the corresponding mode frequency also increases, and the coupling efficiency decreases. Therefore, we should run the simulation a sufficient number of time steps to reach the steady state of on-resonance.

Here we simulate the on-resonance with a mode, $m = 26$, and the gap distance, $g = 0.14\mu m$. When a resonant frequency signal of the ring is continuously launched from the port A of WG1 (Figure 5.2), the signal coupled into the ring will accumulate inside the ring because of the resonance. Some amount of the coupled signal inside the ring will also interact with WG2 and exit from its left port. At the beginning of the simulation, the input signal dominates over the output signal. As the signal inside the ring becomes intensive, then the output signal increases into the amount of input one. After sufficiently long time of simulation, when the resonator reaches the steady state, we can observe almost 100% of the input signal switched into the ring and exit from WG2.

For an off-resonance frequency, the signal coupled into the ring cannot accumulate and become intensive inside the ring, since the frequency does not match with the resonant frequency, thus few signals could be found from the port C of WG2. See Figure 5.12 for both On-resonance and Off-resonance behavior of the ring resonator.

5.4 Conclusion

We studied AWCM and verified its applicability in the area of numerical solutions to the time domain Maxwell's equations with the simulation of micro-ring resonators.

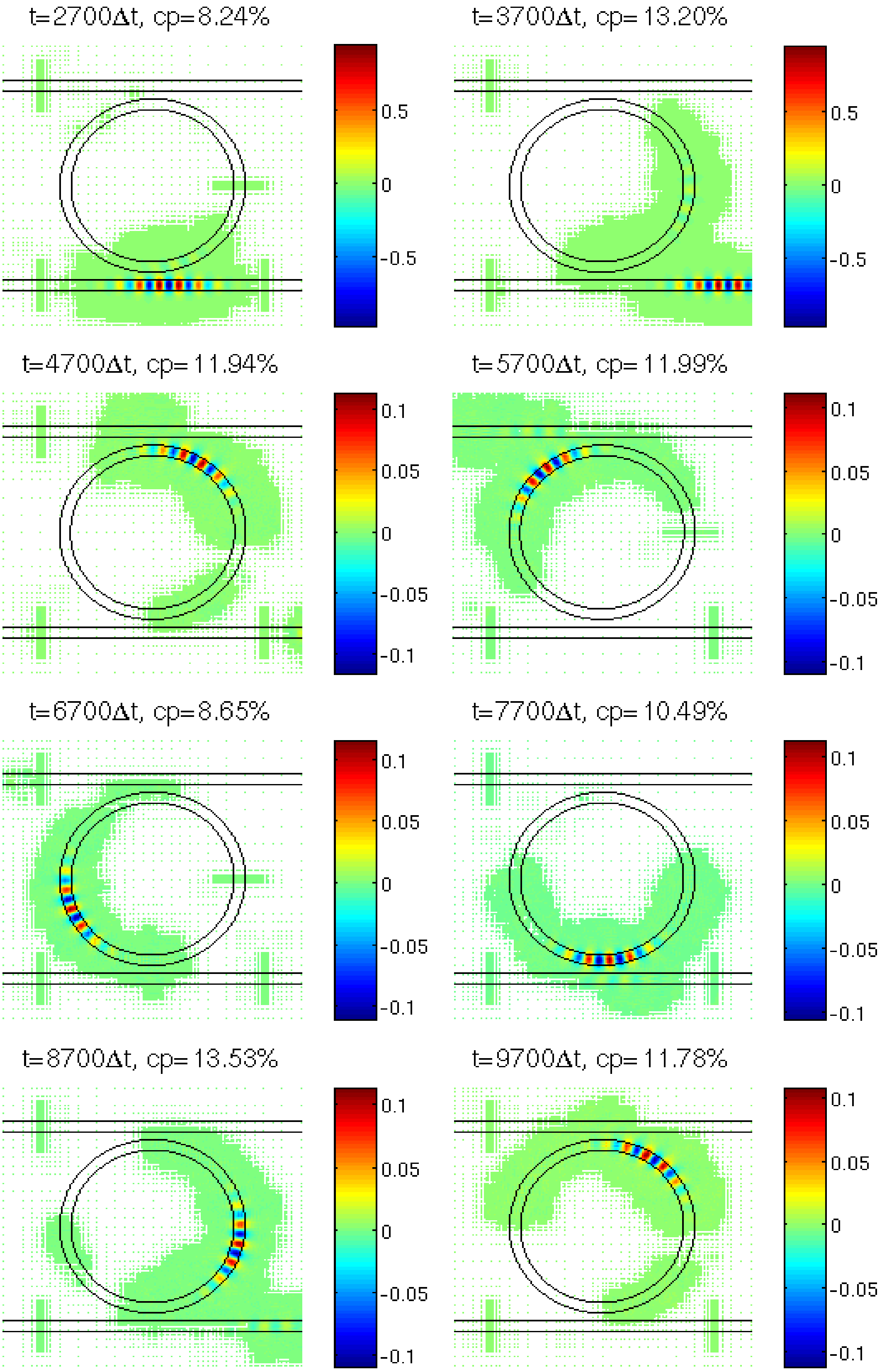
Adapting the grid points dynamically with time stepping is the key feature of the method. The adaptivity property of AWCM enables less storage of data. In each time step, we skip computation on any point in the finest resolution level which is not in the adaptive grid. For those points which are not in the adaptive grid, we only perform an "if" command to check whether that point is included in the adaptive grid or not. The computation of AWCM is mainly done on the points which are in the adaptive grid. The time of computation depends on the percentage of adaptive grid points over the full number of grid points. Since we perform more computations on adaptivity, such as forward and inverse wavelet transforms, interpolations of points needed in the algorithm, than the full grid method, which is non-adaptive, the computation speed is highly effective only when the percentage of the adaptive grid points are very low. The adaptive grid method becomes faster than the full grid method when the percentage of the adaptive grid points are lower than roughly 2.5%. For example, in the simulation of 2D Gaussian peak propagation in the homogeneous media, the adaptive grid method is faster than the full grid method. And also, in the simulation of ring resonators, the percentage of the adaptive grid points becomes close to 2.5% from about the time step 2^{17}, which is half of the number of the total simulation time steps.

According to our experience, the computation on an adaptive grid, whose percentage of adaptive points is about 10%, is still efficient, although the speed of computation is a little bit slower than that of the full grid method. Because we can save a huge amount of the memory space for the storage of output data for visualization. In the AWCM-simulation of ring resonators, the percentage of adaptive grid points is less than 10% for most of the time steps and even less than 2.5% for the second half of the whole simulation time steps.

AWCM is an efficient method for the problems of signals guided by optical waveguides. It will save huge storage of output data for visualization. Especially when signals of interests are highly concentrated inside optical devices so that the compression of grid points for the whole field profile is larger than about 97.5%, AWCM is even faster than the full grid method.

One of the disadvantages of AWCM is slow speed of computation when the compression rate of grid points is low. We are interested in improving AWCM by performing computations on wavelet domain only so that huge amount of computations are saved, since we don't need to do forward and inverse wavelet transforms at each time step to restore physical fields and we even don't need to perform those interpolations procedures in the algorithm.

Another disadvantage of AWCM is on its time integrator. At the moment, we only use second order central difference scheme for discretization of time derivatives in the Maxwell's equations. Thus, no matter how high the order of spatial discretization is, the solution error is restricted by $O(\Delta t^2)$. If a time integrator of higher order is applied in the problem, we may obtain better accuracy with the full grid method so that we can use even smaller threshold tolerance to increase the accuracy of the numerical solution of the adaptive method.


t=2700Δt, cp=8.24%

t=3700Δt, cp=13.20%

t=4700Δt, cp=11.94%

t=5700Δt, cp=11.99%

t=6700Δt, cp=8.65%

t=7700Δt, cp=10.49%

t=8700Δt, cp=13.53%

t=9700Δt, cp=11.78%

Figure 5.4: Snapshots of AWCM-computed $\mathcal{E}_y$ field navigating in the ring resonator. Incident source: a 20-fs Gaussian pulse launched from the left port of the straight waveguide below.

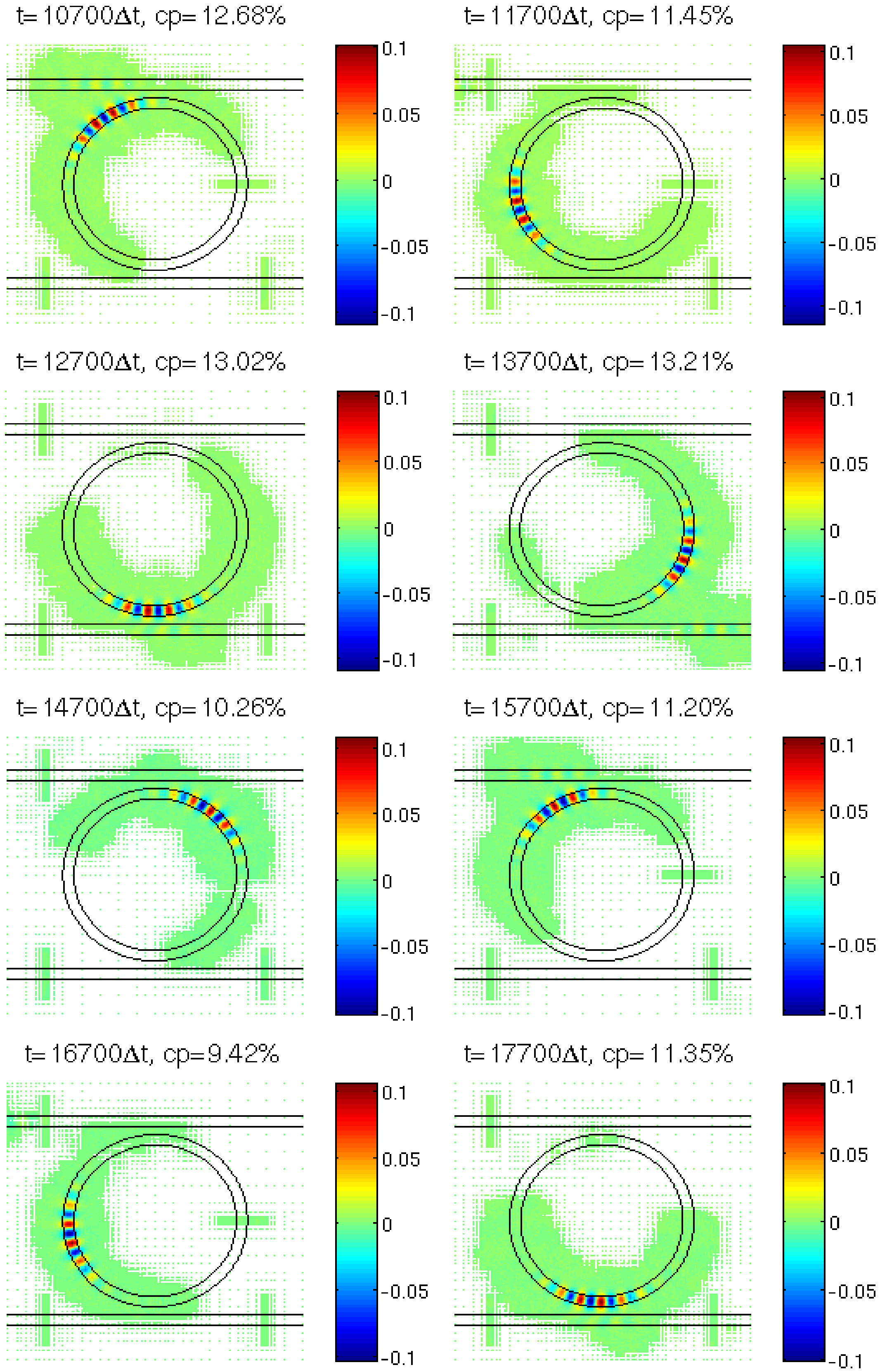
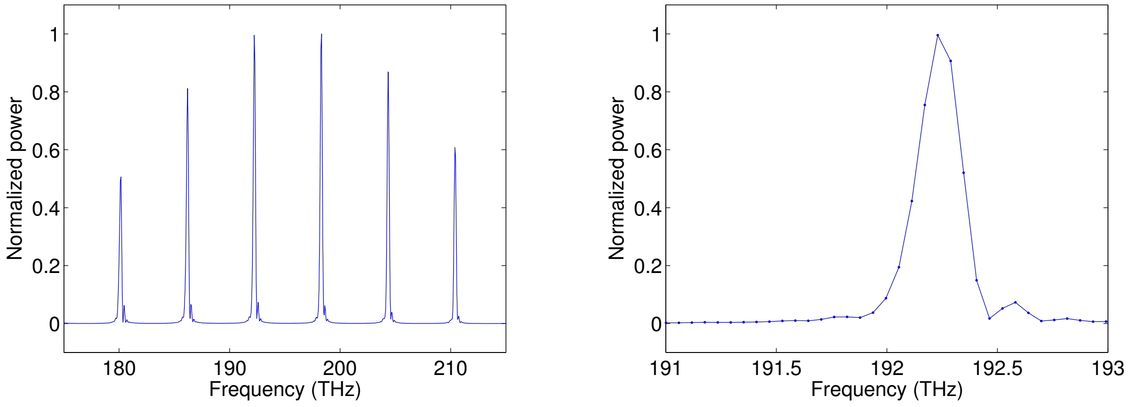


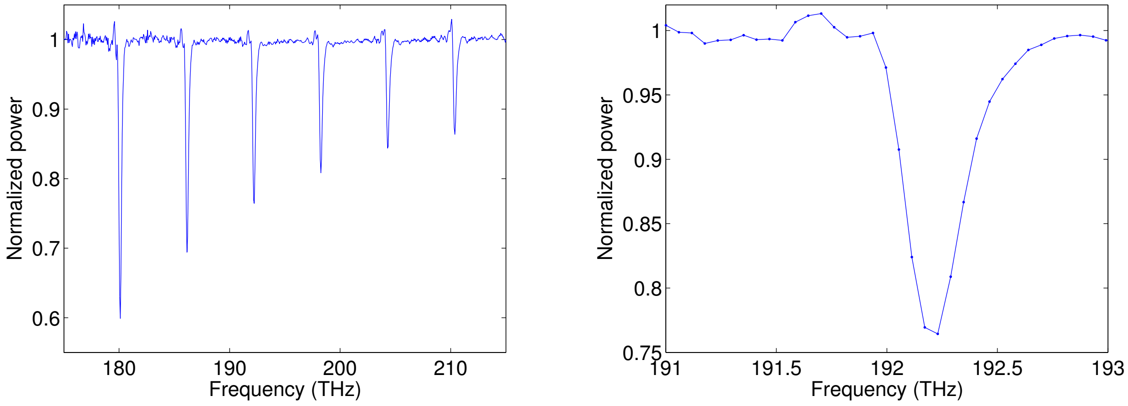
Figure 5.5: More snapshots of the simulation. See the simulation movie in the following link: http://www.youtube.com/user/HaojunLi?feature=mhee#p/u/0/f99PH9tq1VM



(a)

(b)

Figure 5.6: Visualization of the AWCM-computed normalized dropped power. Number of simulation time steps: 2^{18}.



(a)

(b)

Figure 5.7: Visualization of the AWCM-computed normalized transmitted power. Number of simulation time steps: 2^{18}.

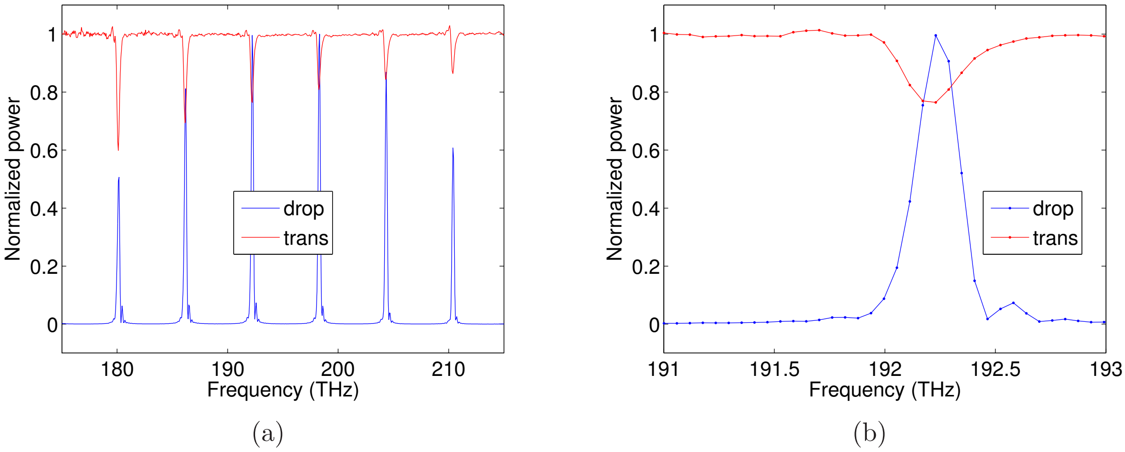


Figure 5.8: Comparison of the AWCM-computed normalized dropped and transmitted power. Number of simulation time steps: 2^{18}.

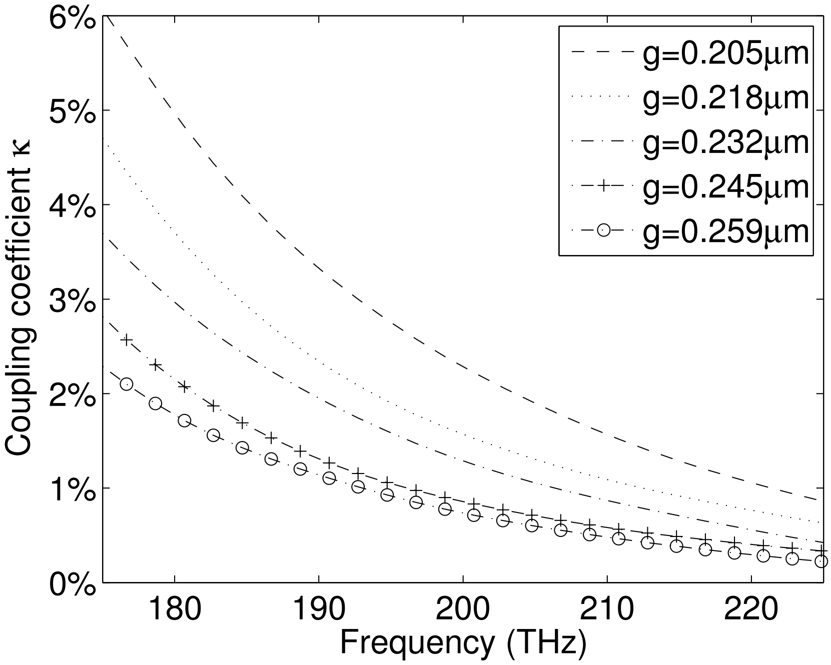
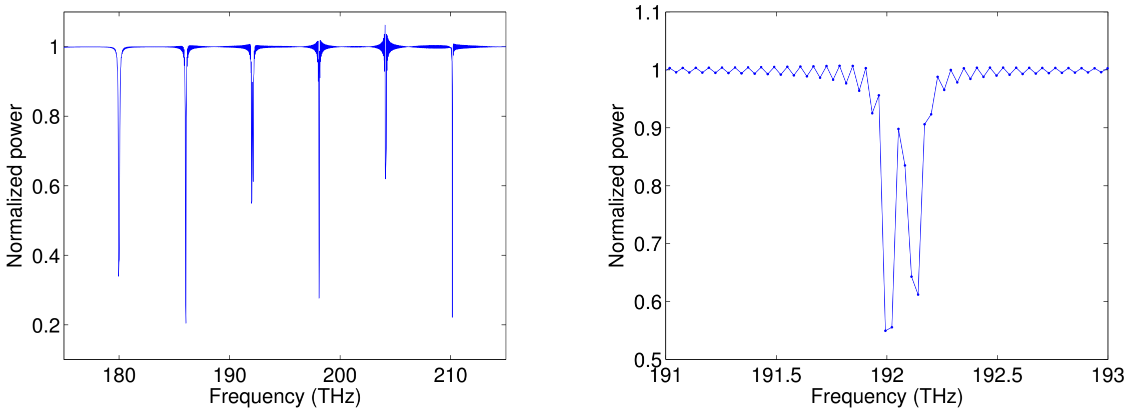
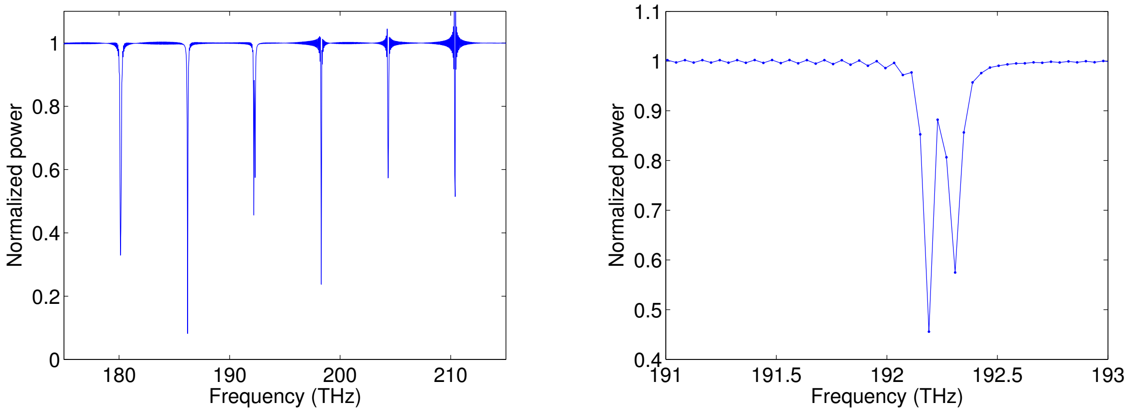


Figure 5.9: Coupling coefficients as a function of frequency and gap distance g for the $d = 0.5\mu m$ resonator.



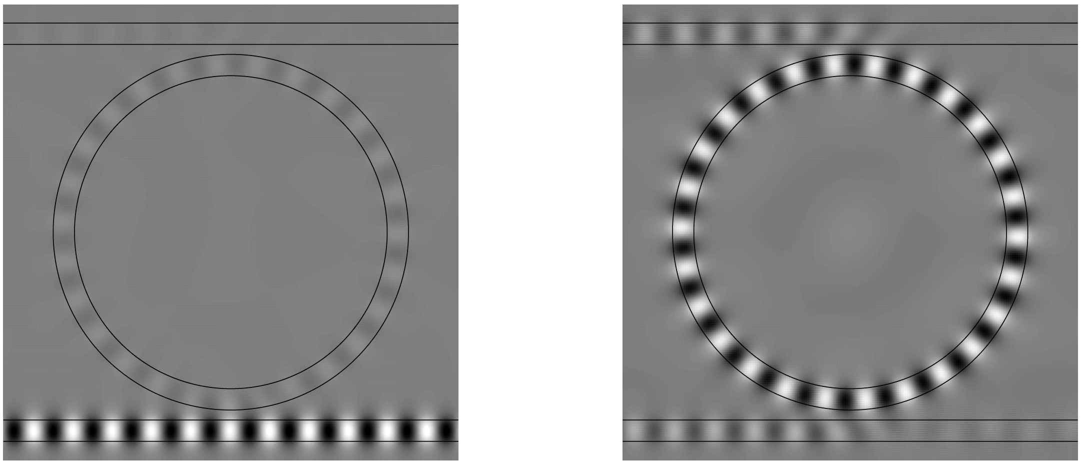
(a) (b)

Figure 5.10: Visualization of the FDTD-computed normalized transmitted power. Simulation time steps: 2^{19}.



(a) (b)

Figure 5.11: Visualization of the ISFM-computed normalized transmitted power. Simulation time steps: 2^{19}.



(a)

(b)

Figure 5.12: Visualization of the AWCM (full grid method) - computed sinusoidal steady state $\mathcal{E}_y$ - field in the $5.0\mu m$ diameter ring with the gap distance $g = 0.14\mu m$. (a) Off-resonance: at 195 THz; (b) On-resonance (m=26): at 192.3 THz.

Appendix

Guided modes of a dielectric slab waveguide

We discuss how to calculate the guided mono-frequency modes of a dielectric slab waveguide, see for example, D. K. Cheng [3]. These modes will be used in our simulations of the steady-state on-off resonance of the ring resonator. And the phasor profiles of these modes will be used in the calculation of a Gaussian pulse modulating a frequency carrier.

Let the permittivity and permeability of the slab waveguide to be ε_1 and μ_1 respectively, and the slab waveguide be surrounded by a vacuum. Assume the width of the slab waveguide to be d and the slab waveguide is symmetric about the x axis. See Figure 5.13.

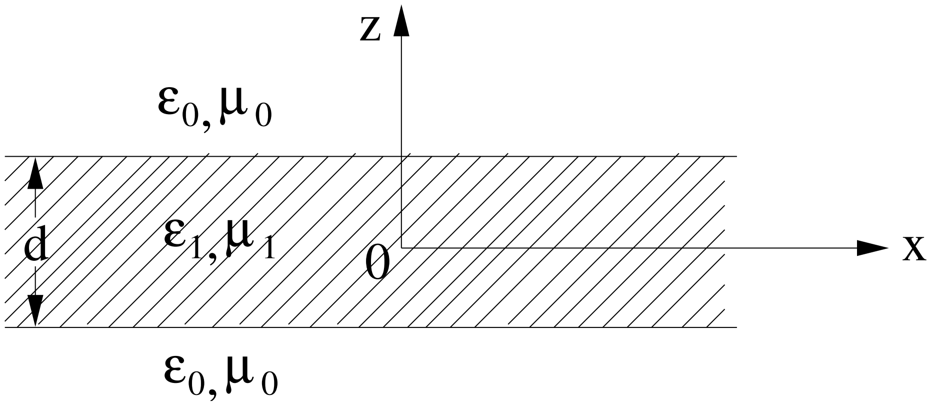


Figure 5.13: A longitudinal cross-section of a dielectric slab waveguide.

We will only consider the guided modes of the slab waveguide for TM$_y$ mode equations (2.10).

Let the time harmonic fields be

$$\mathcal{E}_y(x,z,t) = \Re(\mathbf{E}_y(x,z)\exp(\imath\omega t)), \quad x,z \in \mathbb{R}, t > 0, \tag{5.13a}$$
$$\mathcal{H}_x(x,z,t) = \Re(\mathbf{H}_x(x,z)\exp(\imath\omega t)), \quad x,z \in \mathbb{R}, t > 0, \tag{5.13b}$$
$$\mathcal{H}_z(x,z,t) = \Re(\mathbf{H}_z(x,z)\exp(\imath\omega t)), \quad x,z \in \mathbb{R}, t > 0, \tag{5.13c}$$

where $\mathbf{E}_y$, $\mathbf{H}_x$ and $\mathbf{H}_z$ are *vector phasors* which contain information on directions, magnitudes and phases of the fields. We can write time harmonic TM$_y$ mode equations,

$$\imath\omega\mu\mathbf{H}_x(x,z) =\frac{\partial\mathbf{E}_y(x,z)}{\partial z}, \qquad x,z\in\mathbb{R}, \tag{5.14a}$$

$$\imath\omega\mu\mathbf{H}_z(x,z) =-\frac{\partial\mathbf{E}_y(x,z)}{\partial x}, \qquad x,z\in\mathbb{R}, \tag{5.14b}$$

$$\imath\omega\varepsilon\mathbf{E}_y(x,z) =\frac{\partial\mathbf{H}_x(x,z)}{\partial z}-\frac{\partial\mathbf{H}_z}{\partial x}, \quad x,z\in\mathbb{R}. \tag{5.14c}$$

The guided modes of the slab waveguide are solutions of (5.14) in the following form,

$$\mathbf{E}_y(x,z) =E_y(z)\exp(-\imath\beta x), \quad x,z\in\mathbb{R}, \tag{5.15a}$$

$$\mathbf{H}_x(x,z) =H_x(z)\exp(-\imath\beta x), \quad x,z\in\mathbb{R}, \tag{5.15b}$$

$$\mathbf{H}_z(x,z) =H_z(z)\exp(-\imath\beta x), \quad x,z\in\mathbb{R}, \tag{5.15c}$$

where E_y, H_x and H_z are *profiles* of mode along z direction, and β is the *longitudinal propagation constant.*

If we substitute (5.15) into (5.14), then we get

$$\imath\omega\mu H_x(z) =\frac{dE_y(z)}{dz}, \qquad z\in\mathbb{R}, \tag{5.16a}$$

$$\imath\omega\mu H_z(z) =\imath\beta E_y(z), \qquad z\in\mathbb{R}, \tag{5.16b}$$

$$\imath\omega\varepsilon E_y(z) =\left(\frac{dH_x(z)}{dz}+\imath\beta H_z(z)\right), \quad z\in\mathbb{R}. \tag{5.16c}$$

From (5.16b), we have

$$H_z(z)=\frac{\beta}{\omega\mu}E_y(z), \quad z\in\mathbb{R}. \tag{5.17}$$

We substitute (5.17) into (5.16c), then we get

$$E_y(z)=\frac{\omega\mu}{\imath(\omega^2\varepsilon\mu-\beta^2)}\frac{dH_x(z)}{dz}. \tag{5.18}$$

Now we substitute (5.18) into (5.16a) to obtain

$$h^2H_x+\frac{d^2H_x}{dz^2}=0, \tag{5.19}$$

where $h^2=(\omega^2\varepsilon\mu-\beta^2)$. We know that (5.19) has fundamental general solutions either in the form of cosine and sine terms or in the exponential terms. We require that in the region $|z|\le\frac{d}{2}$, the solution for H_x is

$$H_x(z)=A_1\sin(k_zz)+A_2\cos(k_zz), \quad |z|\le\frac{d}{2}, \tag{5.20}$$

where A_1 and A_2 are constants and

$$k_z^2=\omega^2\varepsilon_1\mu_1-\beta^2. \tag{5.21}$$

If H_x contains only a sine term, we call the mode *odd* and if H_x contains only a cosine term, we call the mode *even.*

In the free space region ($|z| > \frac{d}{2}$), the waves must decay exponentially, thus,

$$H_x(z) = \begin{cases} A_3 \exp(-\alpha(z - d/2)), & z > \dfrac{d}{2}, \\ A_4 \exp(\alpha(z + d/2)), & z < -\dfrac{d}{2}, \end{cases} \tag{5.22}$$

where A_3 and A_4 are constants and

$$\alpha^2 = \beta^2 - \omega^2 \varepsilon_0 \mu_0. \tag{5.23}$$

We use continuity of H_x at the interface $|z| = d/2$ to eliminate the constants A_3 and A_4. For example, for the odd mode case, i.e., $H_x(z) = A_1 \sin k_z z$, we have $A_3 = A_1 \sin(k_z d/2)$ and $A_4 = -A_1 \sin(k_z d/2)$. Using (5.17) and (5.18) we can also obtain corresponding profiles, H_z and E_y.

Here we only list the odd TE modes.

(i) In the dielectric region, $|z| \leq d/2$:

$$H_x(z) = A_1 \sin(k_z z), \tag{5.24a}$$

$$H_z(z) = \frac{\beta}{\imath k_z} A_1 \cos(k_z z), \tag{5.24b}$$

$$E_y(z) = \frac{\omega \mu_1}{\imath k_z} A_1 \cos(k_z z). \tag{5.24c}$$

(ii) In the upper free-space region, $z > d/2$:

$$H_x(z) = A_1 \sin(k_z d/2) \exp(-\alpha(z - d/2)), \tag{5.25a}$$

$$H_z(z) = \frac{\beta}{\imath \alpha} A_1 \sin(k_z d/2) \exp(-\alpha(z - d/2)), \tag{5.25b}$$

$$E_y(z) = \frac{\omega \mu_0}{\imath \alpha} A_1 \sin(k_z d/2) \exp(-\alpha(z - d/2)). \tag{5.25c}$$

(iii) In the lower free-space region, $z < -d/2$:

$$H_x(z) = - A_1 \sin(k_z d/2) \exp(\alpha(z + d/2)), \tag{5.26a}$$

$$H_x(z) = \frac{\beta}{\imath \alpha} A_1 \sin(k_z d/2) \exp(\alpha(z + d/2)), \tag{5.26b}$$

$$E_y(z) = \frac{\omega \mu_0}{\imath \alpha} A_1 \sin(k_z d/2) \exp(\alpha(z + d/2)). \tag{5.26c}$$

Now from (5.24a) and (5.25a), by using continuity of H_x at $z = d/2$, we obtain

$$\frac{\alpha}{k_z} = \frac{\mu_0}{\mu_1} \tan(k_z d/2). \tag{5.27}$$

We also see from (5.21) and (5.23) that

$$\alpha = \sqrt{\omega^2(\varepsilon_1 \mu_1 - \varepsilon_0 \mu_0) - k_z^2}. \tag{5.28}$$

For a given angular frequency ω, we can compute α and k_z by solving the nonlinear equations (5.27) and (5.28). Then, we can obtain β from (5.23).

Bibliography

[1] K. E. Atkinson. *An Introduction to Numerical Analysis*. WILEY, 2nd edition, January 1989.

[2] J. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114:185–200, 1994.

[3] D. K. Cheng. *Field and Wave Electromagnetics*. Addison-Wesley, 2nd edition, January 1989.

[4] C. K. Chui. *An Introduction to Wavelets*. Academic Press, San Diego, 1992.

[5] A. Cohen, I. Daubechies, and J. C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 45, 1992.

[6] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, December 1928.

[7] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 41:909–996, November 1988.

[8] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF Regional Conf. Series in Appl. Math. 61. SIAM, 1992.

[9] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *The Journal of Fourier Analysis and Applications*, 4, 1998.

[10] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constr. Approx.*, 5:49–68, 1989.

[11] S. Dey and R. Mittra. Efficient computation of resonant frequencies and quality factors of cavities via a combination of the finite-difference time-domain technique and the pade approximation. *IEEE, Microwave Guided Wave Lett.*, 8:415–417, 1998.

[12] D. L. Donoho. Interpolating wavelet transforms. Technical report, Department of Statistics, Stanford University, 1992.

[13] S. Dubuc. Interpolation through an iterative scheme. *J. Math. Anal. Appl*, 114:185–204, 1986.

[14] M. Fujii and W. J. R. Hoefer. Wavelet formulation of the finite-difference method: Full-vector analysis of optical waveguide junctions. *IEEE Journal of quantum electronics*, 37(8), 2001.

[15] S. D. Gedney. An anisotropic perfectly matched layer-absorbing medium for the truncation of fdtd lattices. *IEEE, Transactions on antennas and propagation*, 44(12), 1996.

[16] S. C. Hagness, D. Rafizadeh, S. T. Ho, and A. Taflove. Fdtd microcavity simulations: Design and experimental realization of waveguide-coupled single-mode ring and whispering-gallery-mode disk resonators. *Journal of Lightwave technology*, 15(11), 1997.

[17] K. R. Hiremath. *Coupled Mode Theory Based Modeling And Analysis Of Circular Optical Microresonators.* PhD thesis, University of Twente, 2005.

[18] X. Ji, T. Lu, W. Cai, and P. Zhang. Discontinuous galerkin time domain (dgtd) methods for the study of 2-d waveguide-coupled microring resonators. *Journal of Lightwave technology*, 23(11), November 2005.

[19] S. G. Johnson. *http://ab-initio.mit.edu/wiki/index.php/Harminv.*

[20] N. K.-R. Kevlahan and O. V. Vasilyev. An adaptive wavelet collocation method for fluid-structure interaction at high reynolds. *SIAM J. Sci. Comput.*, 26(6):1894–1915, 2005.

[21] D. R. Kincaid and E. W. Cheney. *Numerical analysis.* mathematics of scientific computing. Brooks Cole, 3rd edition, October 2001.

[22] J. Liandrat and P. Tchamitchian. Resolution of the 1d regularized burgers equation using a spatial wavelet approximation. Technical Report 90-83, NASA Langley Research Center, Hampton, VA, 23665-5225, December 1990.

[23] A. K. Louis, P. Maass, and A. Rieder. *Wavelets. Theorie und Anwendungen.* Studienbücher Mathematik, B. G. Teubner, Stuttgart, Germany, 2nd edition, January 1998.

[24] S. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 315:69–87, September 1989.

[25] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 2nd edition, 1998.

[26] V. A. Mandelshtam and H. S. Taylor. Harmonic inversion of time signals and its applications. *J. Chem. Phys.*, 107(17):6756–6769, 1997.

[27] Y. Meyer. *Wavelets and Operators.* Advanced mathematics. Cambridge University Press, 1992.

[28] J. Niegemann, W. Pernice, and K. Busch. Simulation of optical resonators using dgtd and fdtd. *Journal Of Optics A: Pure And Applied Optics*, 11, September 2009.

[29] J. A. Pereda, L. A. Vielva, and A. Prieto. Computation of resonant frequencies and quality factors of open dielectric resonators by a combination of the finite-difference time-domain (fdtd) and prony's methods. *IEEE, Microwave Guided Wave Lett.*, 2:431–433, 1992.

[30] J. D. Regele and O. V. Vasilyev. An adaptive wavelet-collocation method for shock computations. *International Journal of Computational Fluid Dynamics*, 23(7):503–518, August 2009.

[31] N. Satio and G. Beylkin. Multiresolution representations using the autocorrelation functions of compactly supported wavelets. *IEEE Transactions on Signal Processing*, 41(12):319–338, December 1993.

[32] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied And Computational Harmonic Analysis*, 3:186–200, 1996.

[33] W. Sweldens. The lifting scheme: A consctruction of second generation wavelets. *SIAM, J. Math. Anal*, 29(2):511–546, March 1998.

[34] A. Taflove and M. E. Brodwin. Numerical solution of steady-state electromagnetic scattering problems using the time-dependent maxwell's equations. *IEEE, Transactions on Microwave Theory and Techniques*, MTT-23(8):623–630, August 1975.

[35] A. Taflove and S. C. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*. Artech House, 3rd edition, 2005.

[36] E. M. Tentzeris, R. L. Robertson, J. F. Harvey, and L. P. B. Katehi. Stability and dispersion analysis of battle-lemarie-based mrtd schemes. *IEEE, Transactions on Microwave Theory and Techniques*, 47(7):1004–1013, July 1999.

[37] L. N. Trefethen. *Spectral Methods in MATLAB*. SIAM, February 2001.

[38] K. R. Umashankar and A. Taflove. A novel method to analyze electromagnetic scattering of complex objects. *IEEE, Trans. Electromagn. Compat.*, 24:397–405, 1982.

[39] O. V. Vasilyev. Solving multi-dimensional evolution problems with localized structures using second generation wavelets. *International Journal of Computational Fluid Dynamics*, 17(2):151–168, 2003.

[40] O. V. Vasilyev and C. Bowman. Second-generation wavelet collocation method for the solution of partial differential equations. *Journal of Computational Physics*, 165:660–693, 2000.

[41] O. V. Vasilyev and S. Paolucci. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comp. Phys.*, 125:498–512, 1996.

[42] O. V. Vasilyev and S. Paolucci. A fast adaptive wavelet collocation algorithm for multidimensional pdes. *J. Comp. Phys.*, 138:16–56, 1997.

[43] K. S. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE, Transactions on antennas and propagation*, Ap - 14:302–307, 1966.