T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

FINITE DIFFERENCE TIME DOMAIN SIMULATION OF PLANAR WAVEGUIDES

by
Phillip Forkner

A thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science
in the Discipline of Physics

The Graduate School
The University of Tulsa

2016

T H E U N I V E R S I T Y O F T U L S A

THE GRADUATE SCHOOL

FINITE DIFFERENCE TIME DOMAIN

SIMULATION OF PLANAR WAVEGUIDES

by

Phillip Forkner

A THESIS

APPROVED FOR THE DISCIPLINE OF

PHYSICS

By Thesis Committee

_____, Advisor

Dr. Scott Holmstrom

_____

Dr. George Miller

_____

Dr. Peter LoPresti

ABSTRACT

Phillip Forkner (Master of Science in Physics)

Finite Difference Time Domain Simulation of Planar Waveguides

Directed by Dr. Scott Holmstrom

67 pp., Chapter 33:

(206 words)

The main objective of this work is to compute the facet reflectivities of rib waveguides. In photonics, a waveguide is anything which guides light from one point to another in a photonic circuit. A fiber optic cable is one example of a waveguide – a rib waveguide is another. As the light moves through the circuit, it encounters facets, or boundaries, between the devices where some light is reflected and some is transmitted. Computing these reflectivities from first principles is very difficult, if not impossible, in most cases so numerical methods are oftentimes utilized. One such method is the Finite Difference Time Domain (FDTD) method which approximates Maxwell's equations and iterates them over time. I used an implementation of FDTD developed at the Massachusettes Institute of Technology (MIT) called MIT Electromagnetic Equation Propagation (Meep). In this thesis, I will first briefly review Maxwell's equations and how to approximate them with FDTD. Then, I will describe how they can be used to analyze the simpler geometry of planar waveguides. Finally, I will review how Meep was used to obtain the reflectivities of rib waveguides.

ACKNOWLEDGEMENTS

I would like to express sincere thanks to my advisor, Dr. Scott Holmstrom. The completion of this thesis was made possible only through his continuous help and encouragement even when I did not deserve it. Thank you Dr. Holmstrom for your benevolent grace. I would also like to thank the people of the physics department and, specifically, its chair, Dr. George Miller. The TU physics department is a great place to study and to teach and this is due in large part to Dr. Miller's constant efforts. Furthermore, I extend many thanks to the graduate school, especially to Dr. Richard Redner and Hope Geiger. You are so efficient and expedient and provide so much. Finally, I would like to thank my wife, Dae Castaneda. You made this possible. You made this worth it.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

THEORY

1.1 Introduction

Photonics is a burgeoning field in electromagnetics where *photonic circuits* are used to manipulate light in order to transmit information. This can be compared with electronics where electronic circuits are used to manipulate electrons in order to transmit information. Electronic circuits are composed of electronic devices such as resistors, capacitors, and wires whereas photonic circuits are composed of *photonic devices* such as diodes, lasers, and optical fibers. In both cases, a good understanding of the underlying physics is necessary in order to design the devices and much of this understanding stems from the classical theory of electromagnetism.

Perhaps the most basic electronic device is the wire. It is used to guide an electronic signal from one point to another. In photonics, a *waveguide* is used.

[technically, a wire is a waveguide... Zangwill Ch19 Intro. Waveguides of completely of dielectric matter desirable for optics... so the difference between wire and fiber optic cable is just the material and the frequency of the wave?]

A familiar example of a waveguide is the fiber optic cable. When a wire is used to connect different electronic devices, the electrical signal encounters interfaces, or boundaries, between the devices where, for example, contact potentials occur. In phontonics, as light passes through the circuit, it encounters interfaces between the waveguides and devices that make up the circuit. At each interface, some light is transmitted and some is reflected. Thus, knowing how much light is reflected at a given interface is critical to designing photonic circuits. This characteristic is called the *reflectivity*.

One method of analyzing these devices is to apply classical electromagnetic theory which is generally attributed to James Clerk Maxwell who was first to combine electricity and magnetism into a single mathematical theory. However, using this theory, analytical solutions for the reflectivities are not known except in the simplest of cases. Thus, approximate methods are used. One such method is the Finite Difference Time Domain (FDTD) method which is a computational method that makes approximations to Maxwell's equations. These approximate equations are then iterated over time using a computer and the desired electromagnetic fields are solved for. Once the values of the fields are known, the reflectivities can then be computed.

There are several software packages that implement the FDTD method two of which are Lumerical™and MIT Electromagnetic Equation Propagation (Meep). Lumerical is a powerful, commercial software package that includes many computational electromagnetics tools including FDTD. Its main advantage is that it is easy to begin using and has an intuitive graphical user interface. Its main disadvantage is that it requires users to purchase an expensive software license. Meep is also a powerful software package but is free software under the GNU General Public License. It's main disadvantage is that it does not include a graphical user interface and has a steep learning curve in order to use effectively. All of the FDTD simulations in this thesis were done using Meep. However, some of these results will be compared to published results that were created using Lumerical.

For the rest of this chapter, I will briefly go over the classical theory of electrodynamics and introduce some basic concepts that will help in understanding FDTD and the simulations performed in this work. The next chapter will be dedicated to the ideas and approximations of FDTD. Finally in chapter 3, I will introduce Meep and describe some steps necessary to install and use it and present results from my simulations using Meep. Full computer program listings are provided in the Appendix.

In this section, I will give a brief overview of electromagnetism beginning with the microscopic form of Maxwell equations and demonstrating how they lead to the interpretation of light as an electromagnetic wave. Then, after introducing polarization and magnetization,

the Maxwell equations will be writtien in their macroscopic form which is more convenient when dealing with matter. Then simple, non-magnetic matter will be defined and the solutions for the accompanying wave equations will be presented. Boundary conditions will also be described. Then, these equations will be used to analyze symmetric and asymmetric step-index waveguides. Following that, I will describe the effective index method and use it to approximate the effective index of rib waveguides. Finally, I will discuss some other approximation and numerical methods.

1.2 The microscopic Maxwell equations

It is difficult to overestimate the influence James Clerk Maxwell and his unified theory of electricity and magnetism has had on theoretical physics. In his works published during the late 1800's, culminating in the publication of his *Treatise on Electricity and Magnetism* in 1873 [], he proposed 20 equations which brought together all the then-current understanding of electricity and magnetism. Later, Oliver Heaviside, using the vector calculus formalism that he developed, condensed those equations into the four that we use today. We call these equations the "Maxwell equations" or "Maxwell's equations" and their microscopic versions are often written and named as follows:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \qquad \text{Gauss's law,} \tag{1.1a}$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \text{Gauss's law for magnetism,} \tag{1.1b}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \qquad \text{Faraday's law, and} \tag{1.1c}$$

$$\nabla \times \mathbf{B} = \mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \mu_0 \mathbf{j} \qquad \text{Ampere's law.} \tag{1.1d}$$

where $\mathbf{E}$ is the electric field with units of Volts/meter[1] and $\mathbf{B}$ is the magnetic induction with units of Webers/meter^2. Furthermore, ρ is the charge density in units of Coulombs/meter^3 and $\mathbf{j}$ is the current density in units of Amperes/meter^2. Like mass, charge is a fundmental quantity in nature. However, it differs from mass in that it can be positive or negative and

[1]Systeme International (SI) units will be used in this thesis except in the Meep section [].

like charges repel each other while opposite charges attract. A current is simply a moving charge or charges. Finally, $\epsilon_0 = 8.8541878176 \times 10^{-12}$ Farads/meter is the permittivity of free space and $\mu_0 = 4\pi \times 10^{-7}$ Henrys/meter is the permeability of free space. Both are fundamental constants related to the medium they refer to, in this case free space (i.e. the absense of any medium). Roughly speaking, the permittivity is a measure of how easy it is to create an electric field and the permeability a measure of how easy it is to create a magnetic field. Different materials will have different pemittivities and permeabilities.

The $\nabla\cdot$ in the first two equations is the divergence operator and is a measure of how much a quantity diverges or spreads out. In equation (1.1a), Gauss's law, the electric field diverges away from positive charges and towards negative ones. It is typically pictured as arrows pointing away from positive charges and towards negative ones. In equation (1.1b), Gauss's law for magnetism, the magnetic field has zero divergence, i.e. it does not spread out and thus always forms closed loops (this is sometimes referred to as the absence of magnetic monopoles). The next two equations contain the dynamics because they involve the time derivatives. The $\nabla\times$ in both of those equations is the curl operator and is a measure of how much a quantity curls around or loops around. In Faraday's law, equation (1.1c), the electric field loops around a magnetic field that is changing in time. If there is no changing magnetic field, then there is no "loopiness" to the electric field. Finally, in Ampere's law, the magnetic field forms loops around an electric field that changes in time and around electric currents.

Perhaps the greatest shift in thinking that happened during this time, and which Maxwell (and Michael Faraday) introduced, is the idea of "fields of force" pemeating all of space instead of an "action at a distance" as proposed by Isaac Newton. The electric and magnetic fields can exist independently of the charges and currents that created them. Then, the force felt by a charge q moving with velocity $\mathbf{v}$ can be computed from the Lorentz force law:

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \qquad \text{Lorentz force law}$$

It is remarkable that a large part of electromagnetism comes down to manipulating the above five equations. It is also remarkable how compatible this formalism is with more the modern theories of special relativity and quantum mechanics. Indeed, it was Maxwell's equations, and their validity for all inertial frames of reference, that Albert Einstein used as a starting point for his Special Theory of Relativity [4]. Finally, at a time when there were numerous experiments to measure the values of ϵ_0 and μ_0 using charged objects and current-carrying wires, which have little to do with light, Maxwell was able to predict that light is an electromagnetic wave travelling with speed $c = 1/\sqrt{\epsilon_0 \mu_0}$.

1.2.1 Speed of light in vacuum

To see this, first it should be recognized that the Maxwell equations represent a set of coupled partial differential equations. To decouple them, the technique, attributed to Heaviside (Maxwell himself did it differently [9]), is to start with the curl equations (Faraday's and Ampere's law) since they contain the dynamics . Taking the curl of Faraday's law, equation (1.1c), yields,

$$\nabla \times \nabla \times \mathbf{E} + \frac{\partial}{\partial t} \nabla \times \mathbf{B} = 0 \qquad \text{Curl of Faraday's law}$$

where the curl and time derivative of $\mathbf{B}$ were reversed. Then, using the vector identity

$$\nabla \times \nabla \times \mathbf{A} = \nabla \left(\nabla \cdot \mathbf{A} \right) - \boldsymbol{\nabla}^2 \mathbf{A} \qquad \text{Vector identity} \tag{1.2}$$

where $\boldsymbol{\nabla}^2$ is the vector Laplacian, inserting Gauss's law (equation 1.1a), and gathering terms results in the inhomogeneous wave equation for $\mathbf{E}$.

$$\nabla^2 \mathbf{E} - \mu_0 \epsilon_0 \frac{\partial^2 \mathbf{E}}{\partial t^2} = \frac{1}{\epsilon_0} \nabla \rho + \mu_0 \frac{\partial \mathbf{j}}{\partial t} \qquad \text{Inhomogeneous wave equation for } \mathbf{E} \tag{1.3}$$

Following a similar process but starting with Ampere's law yields the inhomogeneous wave

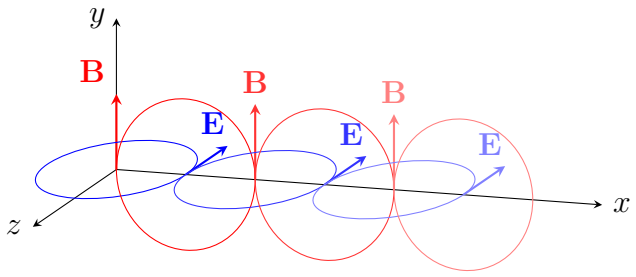


Figure 1.1: In free space ($\rho = \mathbf{j} = 0$), a changing magnetic field (red) creates an encircling electric field (blue) which in turn creates an encircling magnetic field, and so on.

equation for $\mathbf{B}$.

$$\nabla^2\mathbf{B} - \mu_0\epsilon_0\frac{\partial^2\mathbf{B}}{\partial t^2} = -\mu_0\nabla\times\mathbf{j} \qquad \text{Inhomogeneous wave equation for } \mathbf{B} \qquad (1.4)$$

The equations are decoupled but at the price of now having to deal with second order equations. Furthermore, the source terms on the right hand side generally make this problem intractable and thus many methods have been developed to solve them [examples] [Zangwill 715]. However, there are a few simplifications that can readily be made.

The first simplification to the inhomogeneous wave equations (1.3) and (1.4) is to make them homogeneous. In the vacuum of space, there are no charges or currents so $\rho = 0$ and $\mathbf{j} = 0$. In this case, according to the Gauss's laws, both the electric and magnetic fields have no divergence and form closed loops. Furthermore, a changing magnetic field can produce a changing electric field. These changing electric fields can create changing magnetic fields which can create changing electric fields. Maxwell called this the "mutual embrace" of the electric and magnetic fields and it is the basis of how electromagnetic waves propagate through space. Figure 1.1 gives a depiction of this idea. When the speed of these waves is calculated, it turns out to be the speed of light, c. This is one reason that lead Maxwell to believe that light is an electromagnetic wave. Later, Heinrich Hertz experimentally proved the existence of these waves in the form of radio waves.

So, with $\rho = 0$ and $\mathbf{j} = 0$ in the vacuum of space, the inhomogeneous wave equations

become the homogeneous wave equations:

$$\nabla^2\mathbf{E} - \mu_0\epsilon_0\frac{\partial^2\mathbf{E}}{\partial t^2} = 0 \qquad \text{Homogeneous wave equation for } \mathbf{E} \tag{1.5a}$$

$$\nabla^2\mathbf{B} - \mu_0\epsilon_0\frac{\partial^2\mathbf{B}}{\partial t^2} = 0 \qquad \text{Homogeneous wave equation for } \mathbf{B} \tag{1.5b}$$

Thus, the electric and magnetic fields are solutions to the wave equation and travel at the speed of light $c = 1/\sqrt{\epsilon_0\mu_0}$. Incidentally, c stands for the Latin word *celeritas* meaning velocity.

1.3 The macroscopic Maxwell equations

Usually the task when dealing with Maxwell's equations is to solve for the electric and magnetic fields given a particular configuration of charges and currents. The microscopic form the Maxwell equations (1.1) apply equally well to vacuum as to when matter is present however, most of the material physics is tied up in the source terms, ρ and $\mathbf{j}$.

1.3.1 Polarization and displacement

Materials are generally classified as either conductors or insulators. In a conductor, charges are able to move freely through the material whereas in insulators they are not. A *dielectric* is an insulator that can be *polarized*. Even though the electrons and protons that make up the atoms of a dielectric are not able to freely move, whenever it is placed in an electric field they are able to shift where the protons move slightly in the direction of the applied electric field and the electrons move in the opposite direction. This is what is meant by *polarization* since now an otherwise neutral atom or molecule can have a slightly positive charge on one side and negative on the other.

Given that a dielectric can be polarized, it is typical to separate the charge density, ρ, into its bound and free parts, ρ_b and ρ_f. The total charge, then, is the sum of the free

and bound charges,

$$\rho = \rho_f + \rho_b \qquad \text{Total charge}$$

where ρ_f is the free charge density and ρ_b is the bound charge density. In order to get a grasp on what the bound charge is, a polarization, $\mathbf{P}$, is defined as the dipole moment per unit volume which has units of Coulombs/meter^2. Then

$$\rho_b = -\nabla \cdot \mathbf{P} \qquad \text{Bound charge density} \tag{1.6}$$

Note that the polarization points in the same direction as the applied electric field hence the minus sign in above equation. Although the exact form of the polarization will be different for different materials and situations, a larger polarization is the result of a larger charge separation and thus a larger bound charge. Putting the new form for the total charge into Gauss's law and collecting the divergence terms results in

$$\nabla \cdot (\epsilon_0 \mathbf{E} + \mathbf{P}) = \rho_f \qquad \text{Gauss's law}$$

The term in parentheses is defined as the electric displacement,

$$\mathbf{D} \equiv \epsilon_0 \mathbf{E} + \mathbf{P} \qquad \text{Electric displacement field} \tag{1.7}$$

which has units of Coulombs/meter^2.

The polarization can equivalently be written in terms of the *electric susceptibility tensor*, $\boldsymbol{\chi}_e$

$$\mathbf{P} = \epsilon_0 \boldsymbol{\chi}_e \mathbf{E} \qquad \text{Polarization}$$

Now, $\boldsymbol{\chi}_e$ is a dimensionless object and in general can depend on position, time, and the electric and magnetic fields. Plugging this definition of the polarization into the definition

of the electric displacement results in

$$\mathbf{D} = \boldsymbol{\epsilon} \cdot \mathbf{E} \qquad \text{Electric displacement}$$

where $\boldsymbol{\epsilon} \equiv \epsilon_0 (1 + \boldsymbol{\chi}_e)$ is a material parameter called the *permittivity tensor* and will have the same dependence on position, time, etc. as the susceptibility. In the linear regime, $\boldsymbol{\epsilon}$ does not depend on the electric field and thus $\mathbf{D}$ only depends linearly on E (i.e. it does not have any E^2 or higher order dependence). In this case, $\boldsymbol{\epsilon}$ is in general a second rank tensor that can be represented by a matrix. For example, the relationship between $\mathbf{D}$ and $\mathbf{E}$ can be written as

$$\begin{pmatrix} D_1 \\ D_2 \\ D_3 \end{pmatrix} = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix} \begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix}$$

If this matrix is diagonalized, then the diagonal elements (i.e. the eigenvalues) are related to the *principle dielectric constants*. The eigenvectors form the *principle dielectric axes*. When the eigenvalues are all different, the crystal is called biaxial. When two are the same and the third is different, it is called uniaxial (a famous example of a uniaxial material is calcite). Finally, if all the permittivities are the same, the material is isotropic and, in this case, $\mathbf{D}$ and $\mathbf{E}$ are parallel. So, for a linear, isotropic material,

$$\mathbf{D} = \epsilon \mathbf{E} \qquad \text{Displacement field}$$

where ϵ is the (scalar) permittivity of the material. Note that ϵ may still depend on position, time, and frequency.

1.3.2 Magnetization

Similarly, a magnetic material will react to a magnetic field and have an induced magnetization. So, in a similar fashion as the charge density, the current densities are split into bound and free parts. However, there is an additional contribution to the total current

density resulting from changes in the polarization. So the total current density is split into three parts,

$$\mathbf{j} = \mathbf{j}_f + \mathbf{j}_b + \mathbf{j}_p \qquad \text{Total current density}$$

where $\mathbf{j}_f$, $\mathbf{j}_b$, and $\mathbf{j}_p$ are the free, bound, and polarization current densities, respectively. A magnetization, $\mathbf{M}$, which is a magnetic dipole per unit volume having units of Amperes/meter is defined and then

$$\mathbf{j}_b = \nabla \times \mathbf{M} \qquad \text{Bound current density}$$

The current density resulting from changes in polarization is

$$\mathbf{j}_p = \frac{\partial \mathbf{P}}{\partial t} \qquad \text{Polarization current density}$$

Plugging the total current density into Ampere's law and using the definition of the displacement field results in

$$\nabla \times \left(\frac{1}{\mu_0} \mathbf{B} - \mathbf{M} \right) - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{j}_f \qquad \text{Ampere's law}$$

This time, the term in parentheses is defined as the magnetizing field, $\mathbf{H}$,

$$\mathbf{H} \equiv \frac{1}{\mu_0} \mathbf{B} - \mathbf{M} \qquad \text{Magnetic field} \tag{1.8}$$

which has units of Amperes/meter.

Similar statements can be made regarding the magnetization that were made about the polarization. The magnetization is usually written.

$$\mathbf{M} = \boldsymbol{\chi}_m \mathbf{H} \qquad \text{Magnetization}$$

where $\boldsymbol{\chi}_m$ is the magnetic susceptibility tensor. Plugging this into the definition of the magnetizing field results in

$$\mathbf{H} = \boldsymbol{\mu}^{-1}\mathbf{B} \qquad \text{Magnetizing field}$$

where $\boldsymbol{\mu} \equiv \mu_0 (1 + \boldsymbol{\chi}_m)$ is the *permeability tensor*. As with the permittivity, in the linear regime, $\boldsymbol{\mu}$ is a second-rank tensor. If the material is also isotropic then

$$\mathbf{H} = \mu^{-1}\mathbf{B} \qquad \text{Magnetizing field}$$

where μ is the (scalar) permeability of the material.

$\mathbf{D}$ and $\mathbf{H}$ are referred to as the *auxilliary* fields. Putting the definitions for the auxilliary fields (equations 1.7 and 1.8) into the microscopic Maxwell equations results in the macroscopic Maxwell equations:

$$\nabla \cdot \mathbf{D} = \rho_f \qquad \text{Gauss's law} \tag{1.9a}$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \text{Gauss's law for magnetism} \tag{1.9b}$$

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \qquad \text{Faraday's law} \tag{1.9c}$$

$$\nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{j}_f \qquad \text{Ampere's law} \tag{1.9d}$$

Historically, Maxwell introduced $\mathbf{D}$ because he realized that the electric field responsible for the force on a charged particle differed from the field induced in matter by an external charge. Similarly, William Thomson (Lord Kelvin) introduced $\mathbf{H}$ because he intuited that the magnetic field produced by Faraday's law differed in some way from the field induced in matter by a steady external current. Nowadays, the distinction is made between the fields produced by charge and current intrinsic to the matter from those extrinsic to the matter [14] [2.4 p 43].

When light passes from one medium to another, it must pass through the boundary

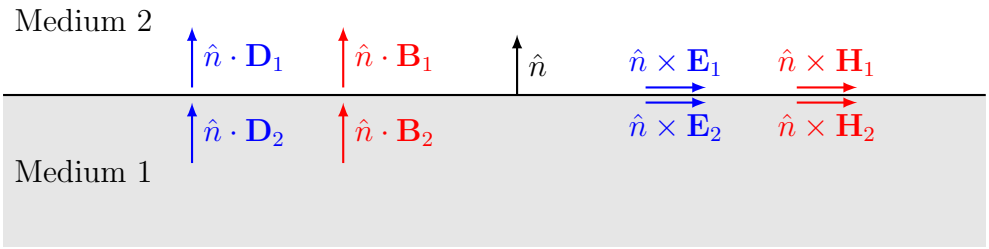


Figure 1.2: The normal components of $\mathbf{D}$ and $\mathbf{B}$ are continuous across the boundary whereas the tangential components of $\mathbf{E}$ and $\mathbf{H}$ are.

between the two media. Some of the light is transmitted and some is reflected. The Maxwell equations impose certain constraints, or *boundary conditions*, on the fields and although it may be easier to understand these conditions from examining the integral form of Maxwell's equations, a step-function can be defined for the fields at the boundary and the differential form of Maxwell's equations 1.9used. Regardless, in a source-free region of space where $\rho_f = \mathbf{j}_f = 0$, they lead to

$$\hat{n} \cdot \mathbf{D}_1 = \hat{n} \cdot \mathbf{D}_2 \qquad \hat{n} \cdot \mathbf{B}_1 = \hat{n} \cdot \mathbf{B}_2 \tag{1.10a}$$

$$\hat{n} \times \mathbf{E}_1 = \hat{n} \times \mathbf{E}_2 \qquad \hat{n} \times \mathbf{H}_1 = \hat{n} \times \mathbf{H}_2 \tag{1.10b}$$

where $\hat{n}$ is a unit normal which points from medium 1 towards medium 2 (see figure 1.2). What these equations say is that the normal components of $\mathbf{D}$ and $\mathbf{B}$ are continuous across the boundary while the tangential components of $\mathbf{E}$ and $\mathbf{H}$ are. Only the normal component of $\mathbf{E}$ or the tangential component of $\mathbf{D}$ can have a discontinuity at the boundary (for $\mathbf{B} = \mu_0\mathbf{H}$) [7, 9].

1.3.3 Speed of light in matter

In summary, for a linear, isotropic material the permittivity and permeability are scalars and the following constitutive relations are valid:

$$\mathbf{D} = \epsilon\mathbf{E} \qquad \mathbf{H} = \mu^{-1}\mathbf{B} \qquad \text{Constitutive relations} \tag{1.11}$$

Putting those relations into the macroscopic Maxwell equations 1.9 results in

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho_f}{\epsilon} \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} &= 0 \\
\nabla \times \mathbf{B} - \mu\epsilon \frac{\partial \mathbf{E}}{\partial t} &= \mu \mathbf{j}_f
\end{aligned}$$

which look exactly like the microscopic Maxwell equations with $\epsilon_0 \to \epsilon$, $\mu_0 \to \mu$, $\rho \to \rho_f$, and $\mathbf{j} \to \mathbf{j}_f$. The inhomogeneous wave equations follow in the same way as (1.3) and (1.4). Finally, in a region where there are no free charges or currents, $\rho_f = \mathbf{j}_f = 0$ which, again, results in the homogeneous wave equations but this time the speed of waves is

$$c = 1/\sqrt{\mu\epsilon} \qquad \text{Speed of light in (simple) matter}$$

During Maxwell's time this was completely sensible since it was believed space was filled with a luminiferous ether through which eletromagnetic waves propagated. So, all one needed to do was change the material parameters, i.e. the permittivity and permeability. After Einstein, who dispensed with the idea of the luminiferous ether, this was no longer tenable. Thus, when an electromagnetic wave enters a material, the changing polarization and magnetization of the material produces electromagnetic waves that travel the same way as in vacuum, just at a different speed [14, 587].

The product $\mu\epsilon$ occurs so often that a dimensionless quantity called the index of refraction is defined as

$$n = \sqrt{\frac{\mu\epsilon}{\mu_0\epsilon_0}} \qquad \text{Index of refraction}$$

It can interpreted as the amount by which an electromagnetic wave slows down in a material.

Furthermore, the ratio μ/ϵ is interpreted as the impedence of the material

$$Z = \sqrt{\frac{\mu}{\epsilon}} \qquad \text{Material impedence}$$

and has units of Ohms. For vacuum $Z = \sqrt{\mu_0/\epsilon_0} \approx 377\,\Omega$.

Writing the macroscopic Maxwell equations only in terms of $\mathbf{E}$ and $\mathbf{H}$ in a source free, non-magnetic, isotropic dielectric where ϵ is independent of time (i.e. structure does not change in time)

$$\begin{aligned} \nabla \cdot \epsilon(\mathbf{r})\mathbf{E} &= 0 \\ \nabla \cdot \mathbf{H} &= 0 \\ \nabla \times \mathbf{E} + \mu_0 \frac{\partial \mathbf{H}}{\partial t} &= 0 \\ \nabla \times \mathbf{H} - \epsilon(\mathbf{r}) \frac{\partial \mathbf{E}}{\partial t} &= 0 \end{aligned}$$

The homogeneous wave equations permit the usual time-harmonic solutions. For example, the electric field can be written in complex form as

$$\mathbf{E}(\mathbf{r}, t) = \boldsymbol{\mathcal{E}}(\mathbf{r}, t) \exp\left[i\left(\mathbf{k} \cdot \mathbf{r} - \omega t\right)\right] \tag{1.12}$$

where $\boldsymbol{\mathcal{E}}(\mathbf{r}, t)$ is the "field envelope" which has position and time dependence. The wave vector $\mathbf{k}$ points in the direction of propagation and has magnitude $k = 2\pi n/\lambda_0$ where λ_0 is the vacuum wavelength. The angular frequency ω is given by $\omega = 2\pi f$ where f is the frequency of the wave. Note that if $\boldsymbol{\mathcal{E}}$ is independent of $\mathbf{r}$ and t, then (1.12) represents a monochromatic plane wave [7, 13]. Taking the real part of (1.12) results in the actual field. The other field quantities can be written in a similar fashion.

"The fields are harmonic, so the Maxwell divergence equations... are satisfied automatically when we impose the Maxwell curl equations,..." [14, p675].

Now, consider a plane wave propagating toward an infinite planar boundary between two dielectric media, 1 and 2. Let the wave be incident from medium 1 and have wave vector $\mathbf{k}_i$. The angle this vector makes with a normal to the surface is called the *angle of incidence* and is denoted by θ_i. Part of the wave is reflected back into medium 1 and its wave vector is denoted by $\mathbf{k}_r$ which makes an angle θ_r with the normal. Finally, the transmitted wave is given the wave vector $\mathbf{k}_t$ and it makes and angle θ_t with the normal, but inside medium 2. The boundary conditions (1.10) require that

$$\mathbf{k}_i \cdot \mathbf{r} = \mathbf{k_r} \cdot \mathbf{r} = \mathbf{k}_t \cdot \mathbf{r}$$

Thus, all three wave vectors lie in the same plane called the *plane of incidence*.

An electromagnetic wave has a *polarization* (not to be confused with polarization of matter discussed above) given by the direction its electric field points. If the electric field is oriented perpendicular to the plane of incidence, then it is called a *transverse electric* (TE) wave. If it is parallel, then the wave is called *transverse magnetic* (TM). The reflection and transmission coefficients for each polarization are obtained by taking the ratio of the reflected and transmitted field amplitudes, respectively, to the incident field amplitude. This results in the so-called Fresnel equations. Furthermore, the *reflectivity* is obtained by taking the ratio of the reflected power to the incident power (the *transmissivity* is then one minus the reflectance), again for each polarization. These equations are all well known and will not be repeated here however, there are two special cases that should be noted. When the angle of incidence is zero, i.e. *normal incidence*, the reflectivity is given by

$$R = \left| \frac{n_1 - n_2}{n_1 + n_2} \right|^2$$

Also, there is a *critical angle* given by

$$\theta_c = \sin^{-1} \frac{n_2}{n_1}$$

If light is internally reflected so that $n_1 > n_2$ then, when light is incident at this angle or larger, no light is transmitted. This phenomenon is called *total internal reflection*.

With these ideas in mind, I will now turn attention to *waveguides*. A waveguide is essentially a device which limits the spatial extent of an electromagnetic wave in one direction and allows it to propagate in the other. The basic structure is an inner guiding *core* surrounded by an outer *cladding* such that the waveguide profile has a constant index in the direction of propagation but not in the transverse direction. When guiding light in the optical part of the spectrum, high-index dielectrics are used for the core and lower index dielectrics for the cladding. This allows for total internal reflection within the waveguide.

Waveguides can be categorized into two basic types: planar and non-planar. A planar waveguide has optical confinement in only one tranverse direction while a non-planar waveguide has confinement in both transverse directions. Fiber optic cables and rib waveguides are an example of non-planar waveguides. Slab waveguides are an example of a planar waveguide. Two slab waveguides that will be discussed here are the symmetric and asymmetric slab waveguides.

Zanwill... beginning of ch19 gives a lot of examples of waveguides.

The slab waveguides considered here consist of three layers: an inner guiding core, an upper cladding, and a lower cladding (see figure 1.3). As light travels down the core, reflections occur at the boundaries with the upper and lower cladding. If the angle of incidence is such that total internal reflection occurs, then the light becomes trapped in the core and propagates down the waveguide with no loss. Furthermore, as the light reflects off of the upper and lower cladding, it interferes with itself - sometimes constructively and sometimes destructively. Thus, there is a *transverse propagation condition* as well as total internal reflection that determines if the light will propagate down the waveguide without loss. Light that meets these conditions are called a *guided modes* and much effort is put into determining what the guided modes are for a particular waveguide geometry.

"A waveguide mode is a transverse field pattern whose amplitude and polarization profiles remain constant along the longitudinal z coordinate" [[7, p74]

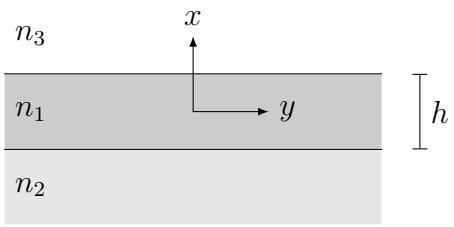


Figure 1.3: Depiction of a slab waveguide.

Zangwill 19.4... conducting tubes... a must read.... and 19.5

CHAPTER 2

FINITE DIFFERENCE TIME DOMAIN (FDTD) METHOD

In this section, I will give a brief overview of the FDTD method, talk about what finite difference approximations are, and then show one example of how those approximations are applied to the governing equations (reference equations from theory section).

2.1 What is the FDTD method?

The FDTD method uses a finite difference approximation to Maxwell's equations to create the so called update equations. These update equations are used to compute the magnetic and electric fields in an iterative, time-stepping fashion. It is a grid based technique so a computational lattice is first setup which includes any objects and materials under study. Usually, all of the fields are initialized to zero then, at the beginning of the simulation, a source (i.e. a current) is "turned on" which creates an electromagnetic field. Then, during the next time step, the magnetic field values are computed at their respective points on the grid. These new values for the magnetic field are then used to update the values for the electric field during the next time step. The new values for the electric field are then used to update the magnetic field and so on, during each consecutive time step. In this way, the fields are computed at every point on the grid at each point in time (i.e. the time domain). This leap-frogging technique is possible because of the unique way in which the magnetic and electric fields are situated on the grid. This configuration is called the Yee cell, discussed below, and allows the computation of the fields from Maxwell's equations directly without having to solve the wave equation.

2.2 Advantages and disadvantages

The finite difference time domain (acronym from Okooi?) method was introduced

in the late 1960s by Kane Yee (yee introduced the cell) and improved upon in the 1970s and 80s. With the advent of more and more powerful computers and their large memory capacities, and new materials manufacturing techniques, FDTD has been gaining popularity as a robust method to simulate various electromagnetic phenomena such as light propagation in photonic crystals, radar simulation for creating absorbing materials, and even simulating the electromagnetic radiation patterns of the entire Earth. It has many advantages over other techniques. For example, the numerical complexity scales linearly with the problem size, whereas with other techniques, it scales exponentially. However, being a grid based technique, the memory requirements still scale as 10^3, and, being a time domain method, simulation times scale as 10^4, emphasizing the need for computers with high processing speeds and large memory capacities. So, another advantage of FDTD is that it is easily parallelized and straighforward to setup on a supercomputer.

Furthermore, whereas other techniques in computational electromagnetics simulate a system at a particular frequency, FDTD can use a broadband pulse source and return the response for multiple frequencies in a single simulation run. Also, given its popularity, there is a plethora of literature on the subject and its errors and limitations have been widely studied. Next, nonlinear materials are naturally included in the simulations through the constitutive relations. Finally, and perhaps most importantly, the FDTD technique is solving the Maxwell equations directly and thus lends itself to visualization of the electric and magnetic fields, field power, etc., which makes it a great learning tool for anyone wanting to better understand electromagnetics.

As with all things, there are disadvantages. For one, it is difficult to incorporate dispersion. Also, being a grid based technique, clever means must be implemented to simulate curved surfaces. Finally, if the device is very small or is resonant, the simulation times can be very long. However, FDTD has been successfully used to simulate things such as light emitting diodes (LEDs), charged coupled devices (CCDs), fiber optics, photovoltaics, and microwave photonics for use in mobile communications.

2.3 The finite difference approximation

At its heart, FDTD is a finite difference method. Since it is the Maxwell curl equations that give the dynamics of the electric and magnetic fields, it is upon those which the finite difference approximations are made. Some examples of ways to approximate the derivative (slope) of a function at the point x, $f'(x)$, are the forward, backward, and central differences.

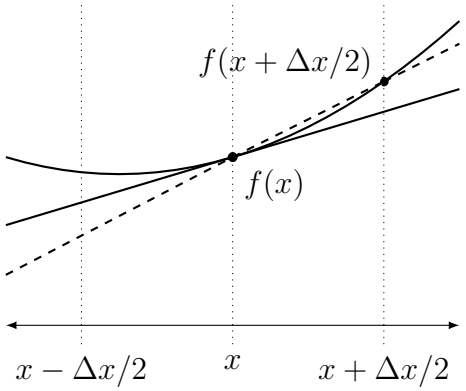
The fundamental theorem of calculus is,

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

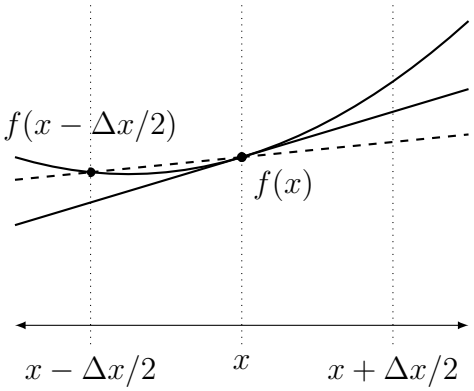
Of course, using a computer, Δx cannot go infinitly to zero. Furthermore, there is a stability criterion relating Δx and Δt. So, what is done is to say

$$\frac{df}{dx} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}.$$

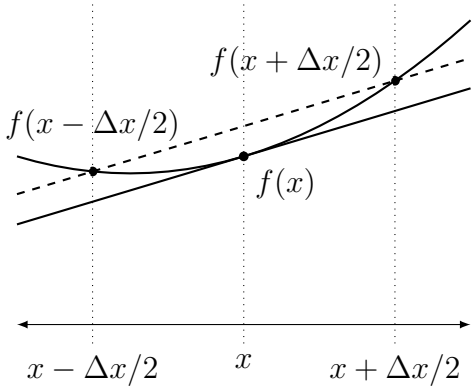
This is, actually, a forward difference approximation. It is approximating the slope of the function f at the point x by evaluating the function at a point $+\Delta x$ in "front" of the point x (see figure 2.1(a)). Similarly, one can appoximate the slope using a point $-\Delta x$ behind the point x. This is called the backward difference. Finally, one can approximate the slope using one point in front of x and one point behind x and then dividing by two. This is the central difference.



(a) Depiction of the forward finite difference approximation.



(b) Depiction of the backward finite difference approximation.



(c) The central difference approximation.

Figure 2.1: Approximations to $f'(x)$. The dashed line represents the approximate slope of $f(x)$. The solid line is the actual slope. Notice how the the central approximation matches the slope most closely in this example.

Taking the central difference interval to be Δx wide, then the difference equations can be written in the following form:

$$f'(x) \approx \frac{f(x + \Delta x/2) - f(x)}{\Delta x/2} \qquad \text{forward difference}$$

$$f'(x) \approx \frac{f(x) - f(x - \Delta x/2)}{\Delta x/2} \qquad \text{backward difference}$$

$$f'(x) \approx \frac{f(x + \Delta x/2) - f(x - \Delta x/2)}{\Delta x} \qquad \text{central difference.}$$

Which of these three approximations is best? An analysis is performed by taking Taylor

expansions of the forward and backward differences:

$$f(x+\Delta x) = f(x) + \frac{\Delta x}{2} f'(x) + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2 f''(x) + \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3 f'''(x) + \cdots$$
$$f(x-\Delta x) = f(x) - \frac{\Delta x}{2} f'(x) + \frac{1}{2!}\left(\frac{\Delta x}{2}\right)^2 f''(x) - \frac{1}{3!}\left(\frac{\Delta x}{2}\right)^3 f'''(x) + \cdots .$$

Solving for $f'(x)$,

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x/2} + \mathcal{O}(\Delta x) \qquad \text{and} \qquad f'(x) = \frac{f(x-\Delta x) - f(x)}{\Delta x/2} + \mathcal{O}(\Delta x).$$

Thus, the forward and backward differences are first-order accurate. Now, if the Taylor expansions are subtracted, then,

$$f'(x) = \frac{f(x+\Delta x/2) - f(x-\Delta x/2)}{\Delta x} + \mathcal{O}\left[(\Delta x)^2\right]$$

which is the central difference. This is second-order accurate. Thus, the central difference approximation is what is usually used in FDTD and what remains is to apply this approximation to the Maxwell equations. There is one more critical step that needs to be taken.

2.4 The Yee cell and the update equations

There are various schemes in which to organize the E and H vectors into a grid context for use on a computer [for example]. However, the Yee cell has proven to be the most robust.

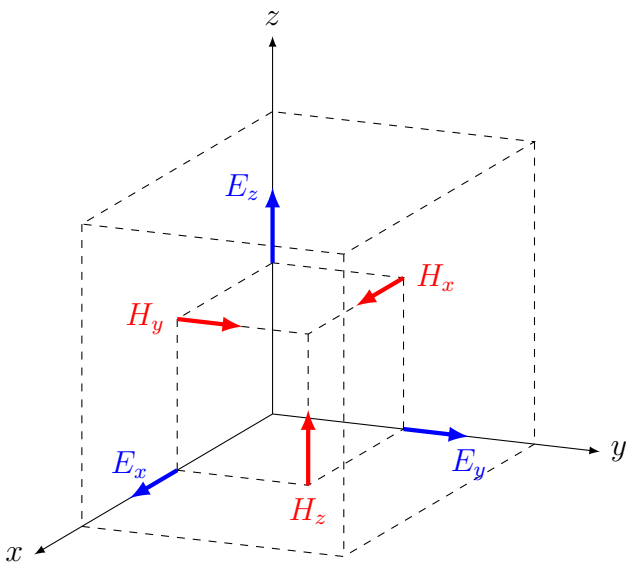


Figure 2.2: The Yee cell

In 1966, Kane Yee introduced a cell geometry (figure 2.2) that elegantly takes into account the divergence equations (the **E** and **H** fields form loops), the boundary conditions ($\mathbf{E}_{\parallel}$ continuous and $\mathbf{H}_{\perp}$ continuous), and provides a clever way to deal with the time derivatives [13]. However, the field components are in different locations and it is possible that they are in different materials (even though they are in the same cell). Additionally, the field components will be out of phase.

Since the divergence equations are automatically satisfied by adopting the Yee cell, the central difference is used on the curl euqations. Furthermore, the **E** and **H** fields are used since they do not include the material parameters. This will allow the creation of the FDTD algorithm independent of the material parameters which can be added at a later time through the use of the constitutive relations. Also, since the electric field is roughly 300 times larger than the magnetic field, it is good practice to first normalize the fields so as not to introduce additional numerical error. Choosing to normalize the magnetic field,

$$\tilde{\mathbf{H}} = \eta_0 \mathbf{H}$$

where the free space impedance is used since the actual impedance is not known in advance (i.e. different materials may be simulatied). Thus, the normalized curl equations are,

$$\nabla \times \mathbf{E} = -\frac{\mu_r}{c_0}\frac{\partial \tilde{\mathbf{H}}}{\partial t} \qquad \text{and} \qquad \nabla \times \tilde{\mathbf{H}} = -\frac{\epsilon_r}{c_0}\frac{\partial \mathbf{E}}{\partial t}.$$

where c_0 is the vacuum speed of light. Applying the central difference approximation to the time derivaties results in,

$$\begin{aligned} \nabla \times E(t) &\approx -\mu \frac{H(t+\Delta t/2) - H(t-\Delta t/2)}{\Delta t}, \text{ and} \\ \nabla \times H(t) &\approx \epsilon \frac{E(t+\Delta t/2) - E(t-\Delta t/2)}{\Delta t}. \end{aligned} \tag{2.1}$$

The problem here is that H exists at $t + \Delta t/2$ in the top equation and at t in the bottom equation (vice-versa for E). To fix this, E is defined to exist at integer multiples of t and H at half-integer multiples, so that the second equation becomes,

$$\nabla \times H(t+\Delta t/2) \simeq \epsilon \frac{E(t+\Delta t) - E(t)}{\Delta t}.$$

Using the top equation in (2.1) to solve for $H(t+\Delta t/2)$ and the above equation to solve for $E(t+\Delta t)$,

$$\begin{aligned} H(t+\Delta t/2) &\approx H(t-\Delta t/2) - \frac{\Delta t}{\mu}\nabla \times E(t) \\ E(t+\Delta t) &\approx E(t) + \frac{\Delta t}{\epsilon}\nabla \times H(t+\Delta t/2). \end{aligned} \tag{2.2}$$

Thus, H at a future time step is determined by using H and E at a previous time step. Then, this new value H is used to compute the new value of E. Hence, the flow of the FDTD algorithm will be: update H from E, then update E from H, and so on.

$\boldsymbol{\mu}$ independent of time.

For the curl equations, expanding them results in,

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{1}{c_0}\left(\mu_{xx}\frac{\partial \tilde{H}_x}{\partial t} + \mu_{xy}\frac{\partial \tilde{H}_y}{\partial t} + \mu_{xz}\frac{\partial \tilde{H}_z}{\partial t}\right)$$
$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{1}{c_0}\left(\mu_{yx}\frac{\partial \tilde{H}_x}{\partial t} + \mu_{yy}\frac{\partial \tilde{H}_y}{\partial t} + \mu_{yz}\frac{\partial \tilde{H}_z}{\partial t}\right)$$
$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{1}{c_0}\left(\mu_{zx}\frac{\partial \tilde{H}_x}{\partial t} + \mu_{zy}\frac{\partial \tilde{H}_y}{\partial t} + \mu_{zz}\frac{\partial \tilde{H}_z}{\partial t}\right)$$

$$\frac{\partial \tilde{H}_z}{\partial y} - \frac{\partial \tilde{H}_y}{\partial z} = -\frac{1}{c_0}\left(\epsilon_{xx}\frac{\partial E_x}{\partial t} + \epsilon_{xy}\frac{\partial E_y}{\partial t} + \epsilon_{xz}\frac{\partial E_z}{\partial t}\right)$$
$$\frac{\partial \tilde{H}_x}{\partial z} - \frac{\partial \tilde{H}_z}{\partial x} = -\frac{1}{c_0}\left(\epsilon_{yx}\frac{\partial E_x}{\partial t} + \epsilon_{yy}\frac{\partial E_y}{\partial t} + \epsilon_{yz}\frac{\partial E_z}{\partial t}\right)$$
$$\frac{\partial \tilde{H}_y}{\partial x} - \frac{\partial \tilde{H}_x}{\partial y} = -\frac{1}{c_0}\left(\epsilon_{zx}\frac{\partial E_x}{\partial t} + \epsilon_{zy}\frac{\partial E_y}{\partial t} + \epsilon_{zz}\frac{\partial E_z}{\partial t}\right)$$

In general, one would apply the central difference approximation to these equations and, as could be imagined, the equations will be very long. However, some assumptions can be made that will simplify things. The interested reader can find the full equations in most texts on FDTD, for instnace in [11].

The first assumption that can be made is that the material is diagonally anisotropic. As mentioned in the theory section, for a diagonally anisotropic material, all of the off-diagonal terms in μ and ϵ are zero. Then, the curl equations become,

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\frac{\mu_{xx}}{c_0}\frac{\partial \tilde{H}_x}{\partial t}, \qquad \frac{\partial \tilde{H}_z}{\partial y} - \frac{\partial \tilde{H}_y}{\partial z} = -\frac{\epsilon_{xx}}{c_0}\frac{\partial E_x}{\partial t},$$
$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\frac{\mu_{yy}}{c_0}\frac{\partial \tilde{H}_y}{\partial t}, \qquad \frac{\partial \tilde{H}_x}{\partial z} - \frac{\partial \tilde{H}_z}{\partial x} = -\frac{\epsilon_{yy}}{c_0}\frac{\partial E_y}{\partial t},$$
$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\frac{\mu_{zz}}{c_0}\frac{\partial \tilde{H}_z}{\partial t}, \quad \text{and} \quad \frac{\partial \tilde{H}_y}{\partial x} - \frac{\partial \tilde{H}_x}{\partial y} = -\frac{\epsilon_{zz}}{c_0}\frac{\partial E_z}{\partial t}.$$

To apply the central difference approximation to these equations, it is common to use superscripts to keep track of the derivatives while reserving the subscripts for the components.

For instance,

$$\frac{\partial E_z}{\partial y} = \frac{E_z^{i,j+1,k}\big|_t - E_z^{i,j,k}\big|_t}{\Delta y}.$$

Note that the E field is being evaluated at time t since the curl term for E in (2.2) was evaluated at t.

Now, since the derivative is being taken with respect to y, the difference is taken between $E_z^{i,j+1,k}$ in the next cell (i.e. $j+1$) and $E_z^{1,j,k}$ in the current cell (i.e j) (see figure 2.3). Furthermore, the material parameters are defined at the same locations as the E and H fields.

Do another example.

Keeping with these conventions, the difference approximations applied to the curl equations results in,

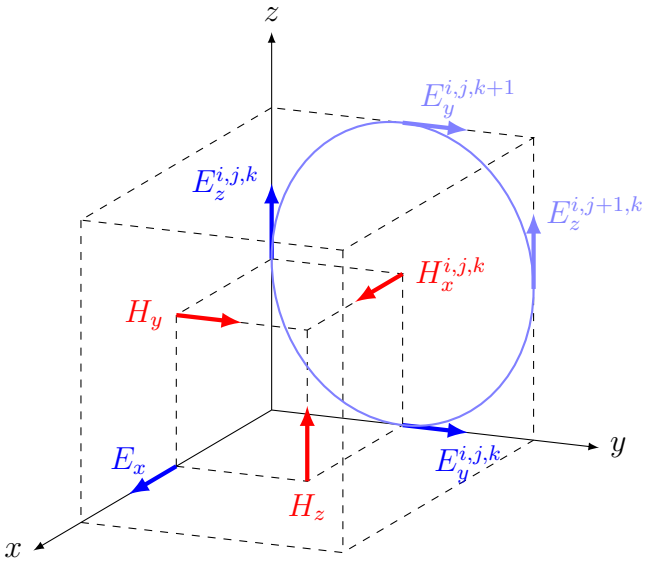


Figure 2.3: Yee cell showing encircling fields around $H_x^{i,j,k}$.

$$\frac{E_z^{i,j+1,k}\big|_t - E_z^{i,j,k}\big|_t}{\Delta y} - \frac{E_y^{i,j,k+1}\big|_t - E_y^{i,j,k}\big|_t}{\Delta z} = -\frac{\mu_{xx}^{i,j,k}}{c_0}\frac{\tilde{H}_x^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{i,j,k}\Big|_{t-\frac{\Delta t}{2}}}{\Delta t},$$
$$\frac{E_x^{i,j,k+1}\big|_t - E_x^{i,j,k}\big|_t}{\Delta z} - \frac{E_z^{i+1,j,k}\big|_t - E_z^{i,j,k}\big|_t}{\Delta x} = -\frac{\mu_{yy}^{i,j,k}}{c_0}\frac{\tilde{H}_y^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_y^{i,j,k}\Big|_{t-\frac{\Delta t}{2}}}{\Delta t},$$
$$\frac{E_y^{i+1,j,k}\big|_t - E_y^{i,j,k}\big|_t}{\Delta x} - \frac{E_x^{i,j+1,k}\big|_t - E_x^{i,j,k}\big|_t}{\Delta y} = -\frac{\mu_{zz}^{i,j,k}}{c_0}\frac{\tilde{H}_z^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_z^{i,j,k}\Big|_{t-\frac{\Delta t}{2}}}{\Delta t},$$

$$\frac{\tilde{H}_z^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_z^{i,j-1,k}\Big|_{t+\frac{\Delta t}{2}}}{\Delta y} - \frac{\tilde{H}_y^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_y^{i,j,k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z} = -\frac{\epsilon_{xx}^{i,j,k}}{c_0}\frac{E_x^{i,j,k}\big|_{t+\Delta t} - E_x^{i,j,k}\big|_t}{\Delta t},$$
$$\frac{\tilde{H}_x^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{i,j,k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z} - \frac{\tilde{H}_z^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_z^{i-1,j,k}\Big|_{t+\frac{\Delta t}{2}}}{\Delta x} = -\frac{\epsilon_{yy}^{i,j,k}}{c_0}\frac{E_y^{i,j,k}\big|_{t+\Delta t} - E_y^{i,j,k}\big|_t}{\Delta t}, \text{ and}$$
$$\frac{\tilde{H}_y^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_y^{i-1,j,k}\Big|_{t+\frac{\Delta t}{2}}}{\Delta x} - \frac{\tilde{H}_x^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{i,j-1,k}\Big|_{t+\frac{\Delta t}{2}}}{\Delta y} = -\frac{\epsilon_{zz}^{i,j,k}}{c_0}\frac{E_z^{i,j,k}\big|_{t+\Delta t} - E_z^{i,j,k}\big|_t}{\Delta t}.$$

At this point, one could solve for the E and H fields at future steps in terms of those values in previous steps. However, another simplifying case is when the system can be represented in one dimension. For example, a layered stack of dielectrics where the layers are very large compared to their thicknesses. Then, in the 1D case with propagation along the z direction, all the derivatives with respect to x and y are zero (since the layers are in the xy plane and considered infinite in extent). Then,

$$-\frac{E_y^{i,j,k+1}\big|_t - E_y^{i,j,k}\big|_t}{\Delta z} = -\frac{\mu_{xx}^{i,j,k}}{c_0}\frac{\tilde{H}_x^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{i,j,k}\Big|_{t-\frac{\Delta t}{2}}}{\Delta t} \tag{2.3a}$$
$$\frac{E_x^{i,j,k+1}\big|_t - E_x^{i,j,k}\big|_t}{\Delta z} = -\frac{\mu_{yy}^{i,j,k}}{c_0}\frac{\tilde{H}_y^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_y^{i,j,k}\Big|_{t-\frac{\Delta t}{2}}}{\Delta t} \tag{2.3b}$$
$$0 = -\frac{\mu_{zz}^{i,j,k}}{c_0}\frac{\tilde{H}_z^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_z^{i,j,k}\Big|_{t-\frac{\Delta t}{2}}}{\Delta t} \tag{2.3c}$$

$$-\frac{\tilde{H}_y^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_y^{i,j,k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z} = -\frac{\epsilon_{xx}^{i,j,k}}{c_0}\frac{E_x^{i,j,k}\big|_{t+\Delta t} - E_x^{i,j,k}\big|_t}{\Delta t} \tag{2.3d}$$

$$\frac{\tilde{H}_x^{i,j,k}\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{i,j,k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z} = -\frac{\epsilon_{yy}^{i,j,k}}{c_0}\frac{E_y^{i,j,k}\big|_{t+\Delta t} - E_y^{i,j,k}\big|_t}{\Delta t} \tag{2.3e}$$

$$0 = -\frac{\epsilon_{zz}^{i,j,k}}{c_0}\frac{E_z^{i,j,k}\big|_{t+\Delta t} - E_z^{i,j,k}\big|_t}{\Delta t} \tag{2.3f}$$

Notice how equation 2.3a and 2.3e both only contain E_y and H_x and how 2.3b and 2.3d only contain E_x and H_y. So, the Maxwell equations decouple into two independent modes. If there is no anisotropy, then the two modes will give the same physics. Looking at the E_y mode, the update equations for the 1D case are,

$$\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} = \tilde{H}_x^k\Big|_{t-\frac{\Delta t}{2}} + \left(\frac{c_0\Delta t}{\mu_{xx}^k}\right)\frac{E_y^{k+1}\big|_t - E_y^k\big|_t}{\Delta z},$$

$$E_y^{i,j,k}\big|_{t+\Delta t} = E_y^{i,j,k}\big|_t + \left(\frac{c_0\Delta t}{\epsilon_{yy}^{i,j,k}}\right)\frac{\tilde{H}_x^k\Big|_{t+\frac{\Delta t}{2}} - \tilde{H}_x^{k-1}\Big|_{t+\frac{\Delta t}{2}}}{\Delta z}.$$

Although these equations look complicated, they can be implemented on a computer with just a few lines of code.[]

With the general idea of how the governing equations are approximated using finite differences and how The overall idea here is that a grid is setup

There are various ways to treat the boundaries of the computational cell. One would be to put a Perfect Electric Conductor (PEC) surrounding your cell in which case the E and H fields would simply be zero. However, sometimes it is necessary to have your computational cell extend to inifinity. In this case, an Absorbing Boundary Condition (ABC) is used. When the electromagnetic wave reaches this boundary, it is simply absorbed. There are varioius techniques to create an ABC but, by far, the most used is the PML [1]. It creates a fictitious material around the computational cell where impedance matching is used to determine the material parameters. This technique is independent of the angle and frequecies of the radiation incident on the boundary [5].

CHAPTER 3

MIT ELECTROMAGNETIC EQUATION PROPAGATION (MEEP)

Now that the basic electromagnetic theory has been covered and the finite difference approximations have been applied to the governing equations, I will decribe an implementation of FDTD called Meep. After giving an overview of Meep, I will go over some of the installation steps to get Meep working on a personal computer and on a supercomputer. Then, I will go over the programming language that is used to run Meep using the program that was used in this work as an example. Finally, I will present some results from the simulations that I ran and compare those to other results.

Meep is a free and open FDTD simulation software package developed at the Massachusetts Institute of Technology (MIT) and licensed under the GNU General Public License (GNU GPL). It has many features including: simultation in 1D, 2D, 3D, and cylindrical coordinates, support for the Message Passing Interface (MPI) standard for parallel computation, Perfectly Matched Layer (PML) boundary conditions, field analysis for flux spectra, among many others [10]. All of the FDTD simulations in this work were done using Meep [11].

3.1 Installation

Meep was developed in a Linux environment, particularly Debian Linux, and it is in a Linux environment that Meep is probably the easiest to install. Most Linux distributions come with a "package manager" that makes it easy to install software from the command line. For example, in Debian (and Ubuntu), it is simply the following one-line command to install everything required.

```
apt-get install meep h5utils
```

In other distributions, you may have to download and install the requirements separately, and there are a qutie a few. However, there are instructions on the Meep webpage and

I have found the Meep discussion email list to be helpful. Of course, you can also download the sources and compile them yourself. This may be useful, for instance, if you plan to run the software on a supercomputer.

Mac OSX is a Unix system. In fact, the terminal runs a Bourne Again SHell (bash) and Meep can also be relatively easily installed from the command line. However, there is no built-in package manager as in Linux, but, you can download and use a package manger called "Homebrew," as mentioned in the Meep installation guide. You will still need to install some prerequisites separately and some packages will need to be compiled, but the process is well documented. You will need to have XCode, Apple's integrated development environment (available for free from the Apple store), installed and have administrator priviledges. As always, you can download the sources and compile them yourself, if you are so inclined.

Now, while Meep can be installed to run natively in a Windows environment, you will need a Linux terminal emulator such as Cygwin and the process is somewhat involved, although it is documented on the interwebs. Perhaps the better way to go is make your machine dual boot, or, run a virtual machine with Windows as the host. Later versions of Windows come with Microsoft's Hyper V virtual machine software and it is easy to setup a Linux distribution of your choice. There are other virtual machine software packages available, notably the open-source VirtualBox provided by Oracle. Beware, running two operating systems at the same time means all resources are being shared. While the CPU load may be minimal with todays advanced processors, it is recommened to have at least 8GB of ram (4GB for Windows and 4GB for the Linux virtual machine) if you are to run any simulations of typical complexity. For instance, the rib waveguides simulated in this work could easily consume 2 GB of ram and sometimes as much as 18 GB.

Finally, the simulations whose results are presented in this thesis were all run on the Tandy Supercomputer (http://www.tandysupercomputing.org). An installation script which outlines the steps I took to get Meep running on the supercomputer is presented in the appendix [].

3.2 The control file and scripting

There are a few ways in which to interface with Meep. Perhaps the preferred way is to write a C++ program that links to the Meep library. After all, Meep was written in C/C++ and this was originally how the package was used. Also, this method gives the greatest level of flexibility in creating FDTD simulations.

Another method is to use a control file, usually abbreviated "ctl" with a name such as `foo.ctl`. The control file is based on libctl (one of the prerequisites installed when Meep was installed) which itself is built on top of the GNU Guile interpreter (also an installed prerequisite) in order to communicate between Scheme, yet another programming language developed at MIT, and other scientific computation software. Although it may seem complicated, the details are not necessary to write a simple ctl file and the Meep website provides some easy to follow tutorials. Once the basic syntax is understood, it is possible to write fairly complex programs (see appendix for the full code listing used in this work).

Editors: emacs, gedit.

A note on units.

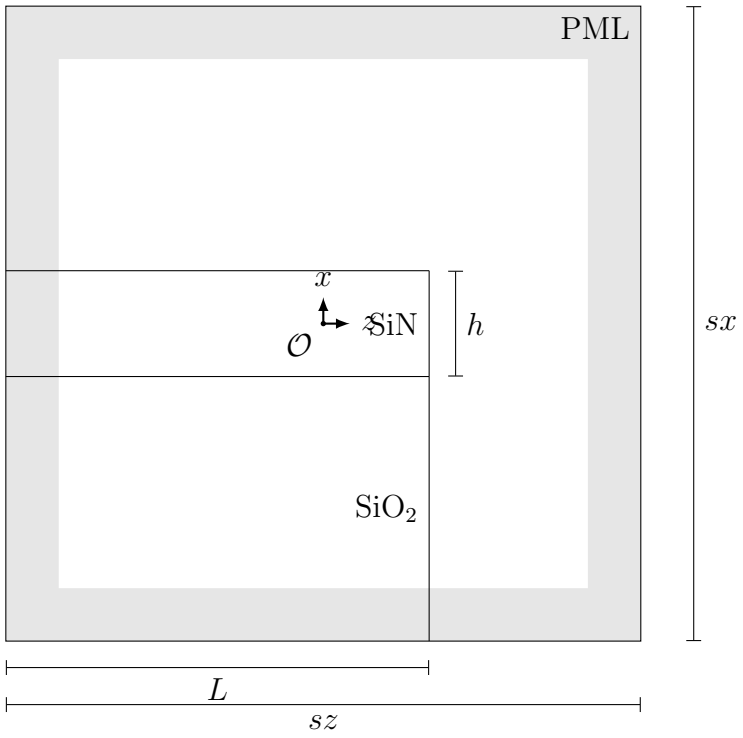


Figure 3.1: A $y = 0$ cross-section of the computaitonal lattice.

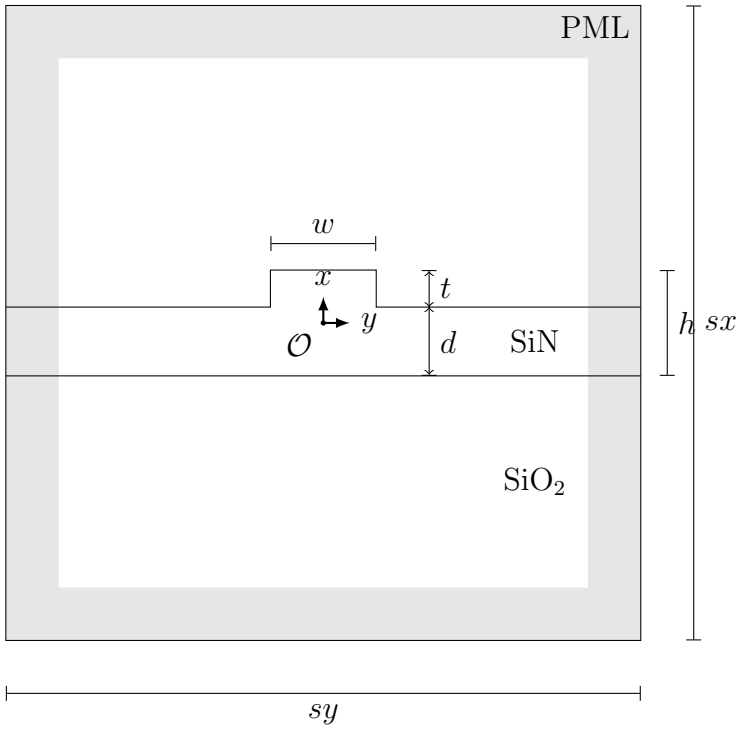


Figure 3.2: A $z = 0$ cross-section of the computational lattice.

I am simulating a rib waveguide (see figures). The propagation direction is z and the x direction is up through the layers. Then, maintaining a right-handed coordinate system, y is horizontally through the core and directed toward the right when looking in the propagation direction.

Mention where the origin is.

As an example, I will go through the code used to simulate the rib waveguide used in this thesis (the complete program listing in provided in the appendix). The first thing to notice is that in libctl, comments are started with the semicolon, `;`, and everything after the semicolon is ignored until a newline is reached. The first section in the .ctl file is

```
; materials
(define SiN (make dielectric (index 1.9827)))
(define SiO2 (make dielectric (index 1.4440)))
(define Si (make dielectric (index 3.4401)))
(define glass (make dielectric (index 1.5)))
```

In libctl, a variable is created with the `(define` *`variable value`*`)` function, where `define` is the libctl function to define a variable, *`variable`* is the name of the variable and *`value`* is the default value. In this case, value takes on object given by `make` which has the syntax

```
(make class (property1 value1) (property2 value2)...)
```

Again, in this case, an object of class type `dielectric` is made. It has one property, `index` with the values as listed. Thus, later in the program when the waveguide is created, these variable names can be used to make the code shorter and more readable.

The next section of code is

```
; waveguide parameters
(define-param h 0.3)
(define hmax 2)
(define L 10)
(define w 2)
(define t (* 0.3 h))
(define d (- h t))
(define spml 0.4)
```

Again, some variables are being defined and their default values set. One difference is the use of `define-param`. Any variable defined this way can be changed from the command line. This makes it very convenient to write a script that will run the simulation for many different values of a parameter, in this case the over height of the core of the waveguide. Please see figures 3.1 and 3.2 for the meanings of h, L, w, t, and d. Another parameter is $h_{\max}$. This is the maximum height that the simulation will run to. Any larger than this, and there will not be enough space between the waveguide and the pml. Of course, the larger $h_{\max}$ is, the larger the computational lattice will be and the longer and larger the simluation will be. The last item to discuss is *spml*. For the most part, anytime there is an "s" in front of a variable name, it stands for "size". Since the material objects in meep are defined by their size and their centers, it is convenient to differentiate between say, x, and sx where x would be the x coordinate of the object and sx would be its size in the x direction. Thus, *spml* is the size of the pml layer. In this case, it is $0.4\,\mu$m all around. Also note that the pml reachings into the lattice and not in addition to it.

The next section sets up the computational lattice.

```
; lattice
(define res 32)
(define sx (* 5 hmax))
(define sy (* 5 w))
(define sz (+ L 1))
(set! geometry-lattice (make lattice (size sx sy sz)))
```

```
(set! pml-layers (list (make pml (thickness spml))))
(set-param! resolution res)
```

Again, there are some `define` statements. Starting at the top, res is the resolution of the simulation. If the computational lattice is a grid, then resolution is how many pieces the grid is broken up into. This is important because the fields are only computed at each grid point. If a boundary or a flux plane, etc. lies between two grid points, then the fields must be interpolated. Thus, a higher resolution will allow for a finer computation of these in between points but, it will also require more memory and a longer compuational time. I ran a simluation, say $10 \times 10 \times 10$ at resolution $res = 3$ and then ran the same simulation as $30 \times 30 \times 30$ at resolution $res = 1$ and got different results.

Moving on, sx, sy, and sz are the size of the computational lattice in the x, y, and z directions, respectively. In the x direction, up through the layers, the size of the lattice is $5h_{\max}$. A couple of things here. First, Meep uses Polish Notation (PN) (also called prefix notation) where the operator is placed to the left of the operands. Thus, `(* 5 hmax)` is interpreted as $5 \times h_{\max}$. The size of the lattice in the x direction is 5 times the maximum height of the waveguide to allow enough space for the wave to propagate away from the waveguide and into the pml. Although I did not use a specific algorithm to come up with the number 5, it is large enough to account for the height of the fundamental mode (i.e. the size of the mode profile in the x direction). Furthermore, the space between the waveguide and the the pml should be at least one wavelength. Since the wavelength used in this simulation was $1.5\,\mu$m, and $h_{\max} = 2\,\mu$m, $sx = 10\,\mu$m was large enough to account for the height of the waveguide and the $0.4\,\mu$m thick pml.

Next, `set!` is used to set pre-defined parameters or objects in Meep. One of the key objects is the computational lattice. The lattice is made with a `make` command and has one property, `size`. Next, the pml is made. The `list` command allows you to create a pml layer for each of the three directions x, y, and z. In this case, I set the the pml to be the same size in all directions. Finally, much like the `define-param` that was used to define custom variables from the command line, `set-param!` allows you to change that

particular pre-defined object from the command line. This was useful when creating the code. For testing purposes, *res* could be set equal to 1 which greatly reduces overhead and computational time.

The next section creates the waveguide.

```
; waveguide
(define-param facet? false) ; no facet for normalization
(define szwvg (if facet? L infinity))
(define zwvg (if facet? (/ (- sz L) -2) 0))
(set! geometry (list
    (make block ; lower cladding
        (size (/ sz 2) infinity szwvg)
        (center (/ sz -4) 0 zwvg)
        (material SiO2)
    )
    (make block ; rib
        (size t w szwvg)
        (center (/ d 2) 0 zwvg)
        (material SiN)
    )
    (make block ; slab
        (size d infinity szwvg)
        (center (/ t -2) 0 zwvg)
        (material SiN)
    )
))
```

Firstly, a boolean variable `facet?` is defined with default value of `false`. Boolean variables in Meep are always followed by a `?`. The reason for defining this boolean variable is that in order to compute the facet reflectivity, a flux plane is created which computes the total flux through that plane. So, if the mode is propagating to the right, goes through the flux plane, is reflected off the facet and propagates to the left, back through the flux plane, then an accounting must be made of how much flux was incident and how much was reflected. This is done by running the simulation twice. In the first run, which is called the normalization run, there is no facet (`facet?=false`), and the total flux through the plane is computed. During the next run, the facet run, there is a facet (`facet?=true`). During this run, the mode propagates to the right, through the flux plane, part of it is reflected off the facet and propagates to the left, back through the flux plane. Meep computes the total flux through this plane for both passes. Thus, the negative of the original, normalization run, is

added to the flux through the plane during the facet run. This way, only the reflected flux is given in the final output. All of this is to say that a boolean variable, `facet` is defined using `define-param` so that it can be changed at the command line.

Next is $szwvg$ which is the size of the waveguide in the z (i.e. propagation) direction. During the normalization run, it is simpy the full z direction of the lattice (actually infinity, but the computational lattice cuts it off, in Meep, it just some huge number, 1×10^{20} I think). During the facet run, it is set to length L which was defined in the waveguide parameters section and, in this case, $L = 10\,\mu$m (in order to compare with Chen).

Another thing to note is the if-then structure used in Meep (Scheme). It goes as follows:

```
(if predicate?  if-true  if-false)
```

So, if `predicate?` is true, then *`if-true`* is executed, otherwise, *`if-false`* is executed. So, in this case, if `facet?=true` then $szwvg = L$, otherwise $szwvg = \infty$.

Similarly, the z coordinate of the waveguide, $zwvg$, is defined with an if-statement. If there is no facet, then the center is simple the center of the computational lattice (i.e. zero). If there is a facet, then $szwvg = L$ and the z coordinate of the waveguide is

$$zwvg = -\frac{1}{2}\left(sz - L\right)$$

Next is setting the geometry of the waveguide. In this case it consists of a series of blocks (i.e. parallelpipeds). Meep supports other geometrical objects such as cylinders, cones, spheres, and ellipsoids. Note that only the intersection of the computational cell and the object is considered in the simulation. The waveguide is comprised of three parts: the lower cladding, the rib, and the slab.

Starting with the lower cladding, it fills the lower half-space of the computational lattice. Thus, its size in the x direction is $sx/2$, in the y direction it is simply ∞ (or, again, some large number). In the z direction, it is the same as $szwvg$. Furthermore, it is centered at $(-sx/4, 0, zwvg)$ and the material is SiO_2. Since this is the first geometrical object to be

defined, any objects which come after (i.e. the rib and slab), will take precedence.

The core of the waveguide is made up of a rib and a slab. Both are made of the same material, SiN. The rib height, t, and the slab height, d, add up to the total height of the core, h. This core is centered at $x = 0$. Thus, the rib center is at $(d/2, 0, zwvg)$ and the slab center is at $(-t/2, 0, zwvg)$. The size of the rib is $(t, w, szwvg)$. The size of the slab is $(d, \infty, szwvg)$. The ∞ in the y size again, is a large number and only the part which intersects the computational lattice is considered during the simulation.

Next comes defining the source.

```
; source
(define wvlen 1.55)
(define srccomp Ey)
(define zsrc (+ (/ sz -2) spml 0.1))
(set! sources (list (make eigenmode-source
    (src (make gaussian-src
        (wavelength wvlen)
        (width 0.5)
    ))
    (component srccomp)
    (size sx (- sy 2) 0)
    (center 0 0 zsrc)
    (direction Z)
)))
```

This section is straightforward in implementation if not in understanding. Indeed, much of my effort went into understanding this particular section of code. The first few items are easy enough, `wvlen` is the vacuum wavelength, in this case $\lambda = 1.55\,\mu$m. Then, the mode component, `srccomp`, is the y component, i.e. the TE mode. Then, $zsrc$ is the z coordinate of the source. In this case it is

$$zsrc = \left(-\frac{sz}{2} + spml + 0.1\right)$$

In other words, it is $0.1\,\mu$m to the right of the pml on the left (negative) side of the lattice. The next part I spent months working on.

Firstly, when the simulation starts, a current source is turned on which creates an electromagnetic wave. After a specified amount of time, the source is turned off and the

wave is allowed to propagate through the simulation region. It is this propagating wave that is being analyzed. In Meep, the source is again created with a `make` statement. In this case, I create an `eigenmode-source`. What this does is make a call to another program created at MIT called MIT Photonic Bands (MPB). MPB uses a planewave expansion method to compute the modes of propagation for a particular waveguide geometry then returns to Meep the current source necessary to excite these modes. So, instead of specifying a generic source, such as a point or line source, and waiting for the simulation to decay into these fundamental modes, MPB speeds up the process by exciting the eigenmode(s) specified.

As can be referred to on the Meep online reference, an eigenmode-source has several properties, five of which are used here. The first is the type of source, specified with the `src` command. In this case, I am using a Gaussian source that has a peak wavelength, *wvlen*, and a width of $0.5\,\mu$m. The next propertry of the eigenmode source is the source component which was defined with the variable *srccomp* corresponding to the TE mode, i.e. the $\mathcal{E}_y$ component. The size of the source is specified in the usual 3 dimensions. For some simulations, I would use a point source, in others I would use a line source. However, when using an eigenmode source, the size of the source is also the computational region sent to MPB. So, this size must be large enough to incorporate all of the features of the waveguide. It should be larger than the width and height of the mode profile.

So, in this case, I used a plane that was the full height of the computational lattice, *sx*, but not quite the full width, *sy* ($2\,\mu$m less actually). For reasons unknown to me, but perhaps related to the pml, MPB did not seem to return the correct fundamental modes for the waveguide if the pml was included in the size of the source. In the x direction, it was not an issue since the guided modes do not exist in the pml there.

There was no z dimension, i.e. the source was a plane.

The size of the source being determined, the location of the center of the source was straightforward. It was centered on the core of the waveguide and put $0.1\,\mu$m from the pml as noted above. Finally, a propagation direction was specified which in this case was the z direction.

There are many other options for the eigenmode source (which I will specify, which are on the Meep reference page) which I experimented with, but these 5 basic settings turned out to be the most important (e.g. the Gaussian source, a planewave source worked equally as well... you have to specify a start and end time... the computation times hardly changed and the results were imperceptably different).

The next section is defining the flux planes. Much time and effort went into getting these just right, but, it turns out, for computing the guided mode reflectivity all you need to do is define the flux plane as a cross-section of the waveguide and then place them appropriately along the length.

```
; flux planes (must be done after geometry, sources, resolution, etc.)
(define zfacet (- L (/ sz 2)))
(define fcen (/ 1 wvlen))
(define df 0)
(define nfreq 1)
(define flux11 (add-flux fcen df nfreq
    (make flux-region ; rib
        (size t w 0)
        (center (/ d 2) 0 (+ zsrc 0.1))
    )
    (make flux-region ; slab
        (size d sy 0)
        (center (/ t -2) 0 (+ zsrc 0.1))
    )
))
(define fluxT1 (add-flux fcen df nfreq
    (make flux-region ; rib
        (size t w 0)
        (center (/ d 2) 0 (+ zfacet 0.1))
    )
    (make flux-region ; slab
        (size d sy 0)
        (center (/ t -2) 0 (+ zfacet 0.1))
    )
))
```

First, I defined $zfacet = L - sz/2$ to be the z coordinate of the facet. This way, a flux plane plane could be easily placed $0.1\,\mu$m away from it. Next is $fcen$. It is the central, or peak, frequency, in this case (and in Meep units) it is simply $1/\lambda$, where λ is the vacuum wavelength (i.e. $\lambda = 1.55\,\mu$m. Next again is df which is the width of frequencies that Meep will consider when computing the fluxes. After that, $nfreq$ is the number of frequencies

within the band df that Meep will compute the fluxes for. So, if $df = 10$ and $nfreq = 10$, then Meep would compute the fluxes for 10 frequencies separated by 1 unit within... One of the advantage of the FDTD method over other methods is that you can compute the fluxes for a band of frequencies in a single simulation. In this work, however, I was only interested in one frequency, $fcen$ and thus $df = 0$ and $nfreq = 1$.

There are two flux planes. One for computing the reflected flux and one for computing the transmitted flux. The reflected flux plane is designated $flux11$ and the transmission flux plane is $fluxT1$. In both cases, the total plane is compsed of two smaller planes: one is the cross-section of the rib and the other is the cross-section of the slab. The rib has a size $(t, w, 0)$ and the slab $(d, sy, 0)$. Unlike when defining the source plane, the flux planes did not change values when the pml was included or not. The only difference between the reflection flux plane and the transmission flux plane is their longitudinal coordinate. The reflection flux plane was placed $0.1\,\mu$m to the right of the source (when looking at the $y = 0$ cross-section in figure 3.1) and the transmission flux plane was located $0.1\,\mu$m to the right of the facet.

I need to describe more the difference between the reflection and transmission flux planes. Could also include a diagram.

The x and y coordinates of the centers correspond to the x and y coordinates of the rib and slab (only the z coordinates change).

Finally, the commands to run the simulation.

```
; ===== RUN THE SIMULATION ==================================================
(use-output-directory (string-append "wvg-out" (number->string h)))
(define decayplus 50)
(define decayval 1e-2)
(define decaypoint (vector3 0 0 (+ zfacet 0.1)))
(if facet? (begin
    (load-minus-flux "flux11-flux" flux11)
))
(run-sources+
    (stop-when-fields-decayed decayplus srccomp decaypoint decayval)
    (at-beginning output-epsilon)
    (at-every 1 (output-png srccomp "-0y 0 -Zc dkbluered -T -S3 -A $EPS -a li
)
(if (not facet?) (begin
    (save-flux "flux11-flux" flux11)
```

```
))
(display-fluxes flux11 fluxT1)
```

The first command, `use-output-directory` does just what it says, it creates an output directory that will be used to store various files output by the simulation. Since I ran the simulations with many values of h, I created an output directory for each. Since h is a number, I used the command `(number->string h)` to convert it to a string. This was then appeneded to `wvg-out` using the `string-append` command.

The next three lines define variables that are used to run the simulation which will be explained below when `run-sources+` is described. As explained above, the simulation is actually run twice. A normalization run is executed first and the computed fluxes are then loaded during the facet run and subtracted from the flux plane. Thus, if `facet?=true`, then the `flux11` fluxes are loaded (negatively). If the .h5 file is not found, Meep will return an error and exit.

Meep provides various `run` functions. The one used here is `run-sources+`. It runs the simulation until a stop condition is reached plus a specified number of additional steps. The stop condition is provided by `stop-when-fields-decayed`. That command takes four arguments: `decayplus` defined with a default value of 50, `srccomp` which was defined in the sources section, `decaypoint` which is a point in the computational lattice, and `decayval` which was defined with default value 1×10^{-2}. At each step, `stop-when-fields-decayed` compares the absolute value squared of the field component specified by `srccomp` at the point `decaypoint` with its value at the beginning of the simulation. If the field has decreased by a factor of `decayval`, then the simulation runs for an additional `decayplus` time steps and stops. So, in this example, when $|\mathcal{E}_y|^2$ at a point 0.1 μm to the right of the facet is 1 hundredth of its initial value, the simulation will run an addition 50 timesteps and stop. All of this is necessary because it is usually not known beforehand how many timesteps will be required for the simulation to run.

I tried values of `devayval` equal to 1/100th and 1/1000th and values of `decayplus` equal to 50 and 100. In any combination of those values, the reflectivities were all within 3

significant figures of each other. Thus, I mostly used the default values set above since they resulted in every so slightly shorter simulation run times.

The next bit of code tells Meep to store the fluxes from the normalization run so they can be loaded during the facet run. Finally, `display-fluxes` outputs the fluxes to stdout.

CHAPTER 4

RESULTS

In this section, I will present the results from simulating three types of waveguides: the symmetric slab waveguide, the asymmetric slab waveguide, and the rib waveguide. The results from the symmetric slab waveguide were compared to published results created by Yijing Chen using Lumerical [2, fig. 7(a)]. In this way it was verified that the Meep code (appendix A.1) was working correctly. Then, the simulation was run again in 3D instead of 2D. Once it was determined that the results were consistent, it was an easy matter to change the indices for each layer and simulate the asymmetric waveguide. Finally, a rib structure was added to the asymmetric waveguide to simulate the rib waveguide. Progressing in this way allowed for minimal change to the code in going from a symmetric slab waveguide, with published results, to the rib waveguide studied in this work.

In all cases, the wavelength of the source was $\lambda = 1.55\,\mu$m and the lengths of the waveguides were $L = 10\,\mu$m. The symmetric waveguide consisted of a silicon core with index of refraction $n_1 = 3.44$ and an upper and lower cladding of silicon dioxide having index of refraction $n_2 = n_3 = 1.44$. The thickness, or height, of the core was simulated for $0.05\,\mu\text{m} \leq h \leq 1\,\mu\text{m}$. The total power reflectivity is plotted versus $V/2\pi$ (or $h/(\lambda/dn)$) in figure (4.1). Chen's results are also plotted as a reference. As can be seen, there is good agreement between Meep and Lumerical with both showing the characteristic increase in reflectivity as the number of supported modes increases from 1 to 2.

The next step was to run the simulation in 3 dimensions. Those results are shown in figure (4.2) along with the 2D results for comparison. As can been seen, there is good agreement between the 2D and 3D simulations although the 3D results are slightly lower than the 2D ones. Then, the indices were changed in order to simulate the asymmetric waveguide.

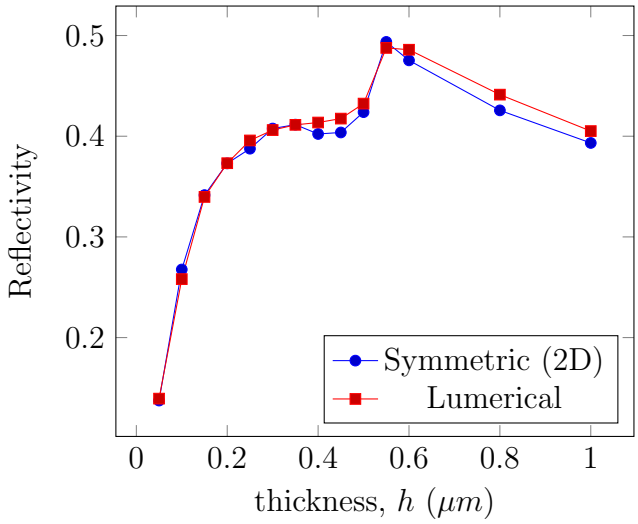


Figure 4.1: A plot showing reflectivity curves for a symmetric waveguide using Meep (2D) and Lumerical (Chen).

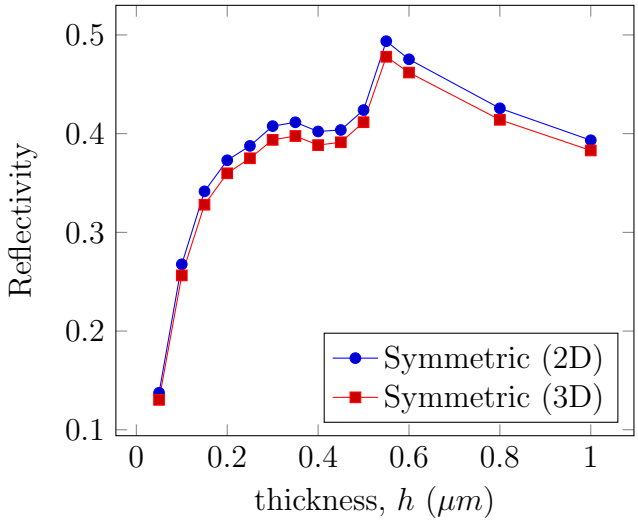


Figure 4.2: A plot showing reflectivity curves for a symmetric waveguide using Meep in 2D and 3D.

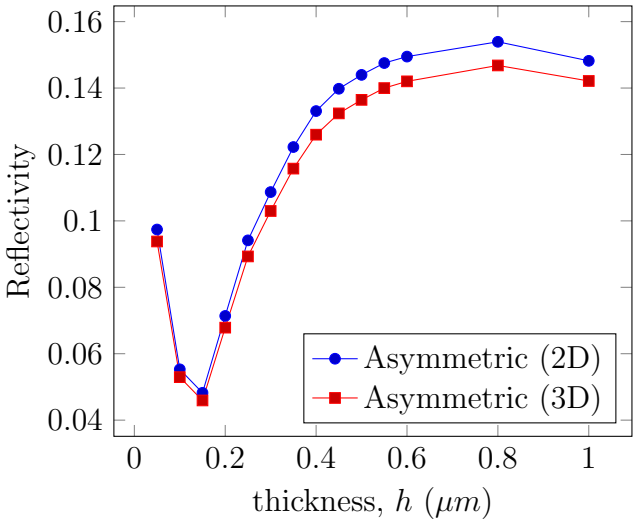


Figure 4.3: A plot showing reflectivity curves for an asymmetric waveguide using Meep in 2D and 3D.

Those results are shown in figure (4.3) . Again, there is good agreement between the two cases and the 3D case is slightly lower than the 2D one. Also shown is the experimental value R_{meas} for a $2\,\mu$m rib waveguide.

In the asymmetric case, there is no sudden increase as in the symmetric case because the waveguide is single mode for all thicknesses. However, the reflectivity does increase as $V/2\pi$ decreases below about 0.1 - corresponding to a thickness of $h = 0.1\,\mu$m. I am not sure if there is a physical reason for this or if it is some kind of a numerical artifact. However, when I tried increasing the resolution of the simulation, the "artifact" remained.

(The simulation was also run out to $2\,\mu$m and began to approach the infinite planar boundary case.)

Next, a $2\,\mu$m rib was added to the asymmetric slab waveguide thus creating the rib waveguide. The rib waveguide is inherently a 3D simulation so the 3D asymmetric reflectivities are shown for comparison in figure (4.4) as well as R_{meas}. As can be seen, for the computed thicknesses, the rib waveguide tends to have a lower reflectivity. This makes sense since the slab thickness of the rib waveguide is smaller than that of the asymmetric waveguide for a given height (recall $h = t + d$ for the rib waveguide). [It might be worthwhile to plot the rib waveguide versus d instead of h.]

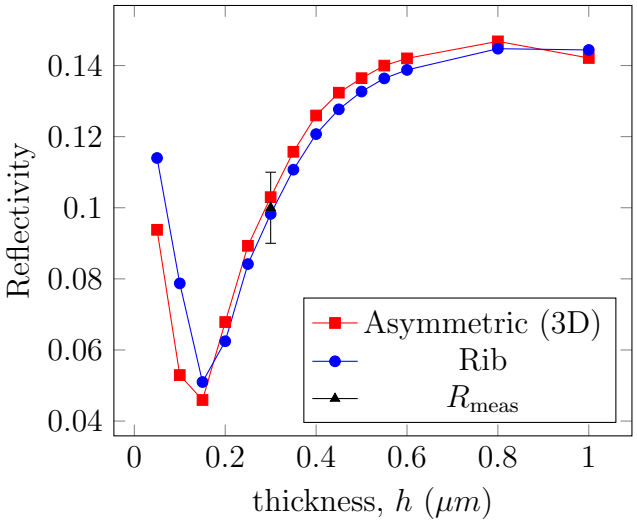


Figure 4.4: A plot showing reflectivity curves for an asymmetric waveguide and rib waveguide.

In figure 4.5, rib waveguide reflectivities are compared with the R_4 model.

Finally, the rib waveguide was simulated with rib widths of $w = 1\,\mu$m and $w = 0.5\,\mu$m. Those reflectivities are shown in figure 4.6 along with the $2\,\mu$m rib waveguide. As can be seen, the reflectivity generally decreases as the rib width decreases. Again, this makes sense because as the rib width becomes smaller and smaller, the waveguide approaches that of asymmetric waveguide.

Can also include some results about memory requirements and simulation times.

TM mode.

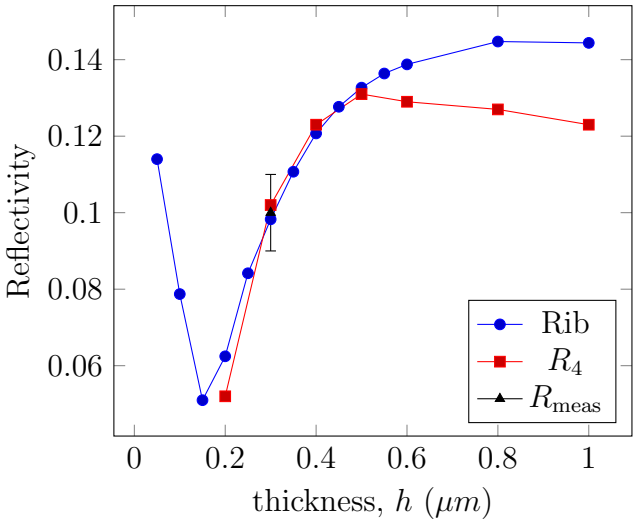


Figure 4.5: Meep versus R_4 versus R_{meas}.

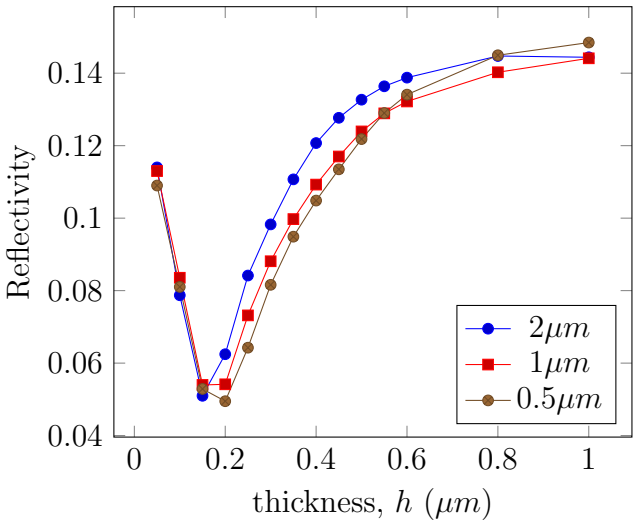


Figure 4.6: A plot showing reflectivity curves for rib waveguides with three different rib widths.

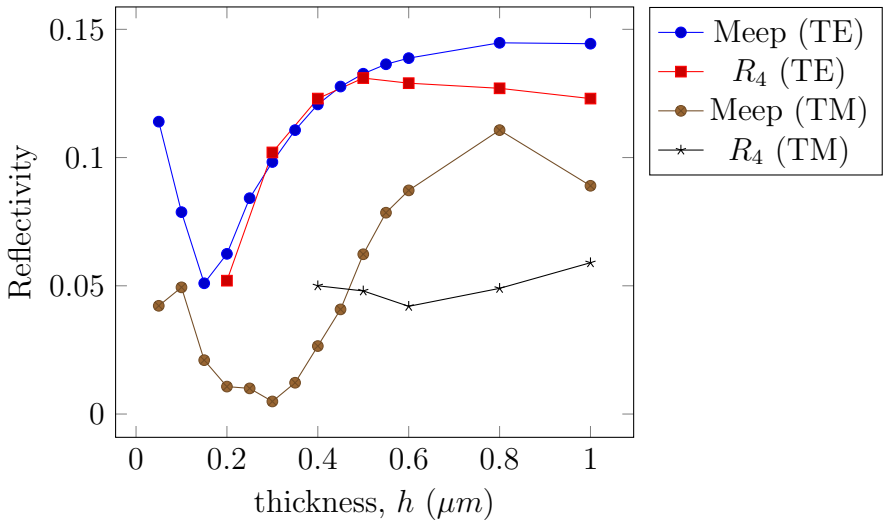


Figure 4.7: TE and TM mode for 2 μm rib waveguide.

BIBLIOGRAPHY

[1] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, Oct 1994.

[2] Yijing Chen, Yicheng Lai, Tow Chong Chong, and Seng-Tiong Ho. Exact solution of facet reflections for guided modes in high-refractive-index-contrast sub-wavelength waveguide via a Fourier analysis and perturbative series summation: Derivation and applications. *Journal of Lightwave Technology*, 30(15):2455–2471, Aug 2012.

[3] Dok Hee Choi and W.J.R. Hoefer. The finite-difference-time-domain method and its application to eigenvalue problems. *IEEE Trans. Microwave Theory Techn.*, 34(12):1464–1470, Dec 1986.

[4] A. Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 322:891–921, 1905.

[5] Atef Z. Elsherbeni and Veysel Demir. *The Finite Difference Time Domain Method for Electromagnetics: With MATLAB Simulations*. SciTech Publishing, 2009.

[6] David J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, 3rd edition, 1999.

[7] J. Liu. *Photonic Devices*. Cambridge University Press, 2009.

[8] E. A. J. Marcatili. Dielectric rectangular waveguide and direction. *The Bell System Technical Journal*, 48:2071–2121, Mar 1969.

[9] James Clerk Maxwell. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–513, 1865.

[10] Ardavan F. Oskooi, David Roundy, Mihai Ibanescu, Peter Bermel, J. D. Joannopoulos, and Steven G. Johnson. MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method. *Computer Physics Communications*, 181:687–702, January 2010.

[11] A. Taflove and S.C. Hagness. *Computational Electrodynamics: The Finite-difference Time-domain Method*. Artech House antennas and propagation library. Artech House, 2005.

[12] Abdorreza Torabi, Amir Ahmad Shishegar, and Reza Faraji-Dana. Analysis of modal reflectivity of optical waveguide end-facets by the characteristic green's function technique. *Journal of Lightwave Technology*, 32(6):1168–1176, Mar 2014.

[13] Kane Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, May 1966.

[14] Andrew Zangwill. *Modern Electrodynamics*. Cambridge University Press, 2013.

APPENDIX A

PROGRAM LISTINGS

A.1 2D symmetric waveguide

```
; Planar waveguide: Si in glass
; 2 dimensions
; Phillip Forkner
; units are microns

;     ----------------------------------------
;    |                                    PML |
;    |     ----------------------------------     |
;    |    |                                  |    |
;    |    |                                  |    |
;    |    | upper           x                |    |
;    |    |                 ^                |    |
;    |    |-----------------|----------------|    |
;    |    |                 |                |    |
;sx  |    | core            ---> y           | h  |
;    |    |                                  |    |
;    |    |----------------------------------|    |
;    |    |                                  |    |
;    |    | lower                            |    |
;    |    |                                  |    |
;    |    |----------------------------------|    |
;    |                                            |
;    |--------------------------------------------|
;                          sy

; materials
(define SiN (make dielectric (index 1.9827)))
(define SiO2 (make dielectric (index 1.4440)))
(define Si (make dielectric (index 3.4401)))
(define glass (make dielectric (index 1.5)))

; waveguide parameters
(define-param h 0.3)
(define hmax 2)
(define L 10)
(define w 2)
(define spml 0.4)

; lattice
(define res 32)
(define sx (* 5 hmax))
(define sy no-size)
(define sz (+ L 1))
(set! geometry-lattice (make lattice (size sx sy sz)))
(set! pml-layers (list (make pml (thickness spml))))
(set-param! resolution res)

; waveguide
(define-param facet? false) ; no facet for normalization
(define szwvg (if facet? L infinity))
(define zwvg (if facet? (/ (- sz L) -2) 0))
```

```
(set! geometry (list
    (make block ; upper cladding
        (size (/ sz 2) infinity szwvg)
        (center (/ sz 4) 0 zwvg)
        (material glass)
    )
    (make block ; lower cladding
        (size (/ sz 2) infinity szwvg)
        (center (/ sz -4) 0 zwvg)
        (material glass)
    )
    (make block ; core
        (size h infinity szwvg)
        (center 0 0 zwvg)
        (material Si)
    )
))

; source
(define wvlen 1.55)
(define srccomp Ey)
(define zsrc (+ (/ sz -2) spml 0.1))
(set! sources (list (make eigenmode-source
    (src (make gaussian-src
        (wavelength wvlen)
        (width 0.5)
    ))
    (component srccomp)
    (size sx 0 0)
    (center 0 0 zsrc)
    (direction Z)
)))

; flux planes (must be done after geometry, sources, resolution, etc.)
(define zfacet (- L (/ sz 2)))
(define fcen (/ 1 wvlen))
(define df 0)
(define nfreq 1)
(define flux11 (add-flux fcen df nfreq
    (make flux-region
        (size h 0 0)
        (center 0 0 (+ zsrc 0.1))
    )
))
(define fluxT1 (add-flux fcen df nfreq
    (make flux-region
        (size h 0 0)
        (center 0 0 (+ zfacet 0.1))
    )
))

; ===== RUN THE SIMULATION ================================================
```

```
(use-output-directory (string-append "wvg-out" (number->string h)))
(define decayplus 100)
(define decayval 1e-2)
(define decaypoint (vector3 0 0 (+ zfacet 0.1)))
(if facet? (begin
    (load-minus-flux "flux11-flux" flux11)
))
(run-sources+
    (stop-when-fields-decayed decayplus srccomp decaypoint decayval)
)
(if (not facet?) (begin
    (save-flux "flux11-flux" flux11)
))
(display-fluxes flux11 fluxT1)
```

A.2 Rib waveguide

```
; Rib waveguide
; 3 dimensions
; Phillip Forkner
; units are microns

;     ------------------------------------------
;    |                                      PML |
;    |     ---------------------------------    |
;    |    |                 w               |   |
;    |    |               -------             |   | _
;    |    |air           |       |            |   | |
;    |    |              |       | t          |   | |
;    |    |______________|       |____________|   | |
;    |    |                                    |   | | h
;sx  |    |core                                | d | |
;    |    |                                    |   | |
;    |    |____________________________________|   | _
;    |    |                                    |   |
;    |    |lower                               |   |
;    |    |                                    |   |
;    |    |____________________________________|   |
;    |                                             |
;    |_____________________________________________|
;                          sy

; materials
(define SiN (make dielectric (index 1.9827)))
(define SiO2 (make dielectric (index 1.4440)))
(define Si (make dielectric (index 3.4401)))
(define glass (make dielectric (index 1.5)))

; waveguide parameters
(define-param h 0.3)
(define hmax 2)
(define L 10)
(define w 2)
(define t (* 0.3 h))
(define d (- h t))
(define spml 0.4)

; lattice
(define res 32)
(define sx (* 5 hmax))
(define sy (* 5 w))
(define sz (+ L 1))
(set! geometry-lattice (make lattice (size sx sy sz)))
(set! pml-layers (list (make pml (thickness spml))))
(set-param! resolution res)

; waveguide
(define-param facet? false) ; no facet for normalization
```

```
(define szwvg (if facet? L infinity))
(define zwvg (if facet? (/ (- sz L) -2) 0))
(set! geometry (list
    (make block ; lower cladding
        (size (/ sz 2) infinity szwvg)
        (center (/ sz -4) 0 zwvg)
        (material SiO2)
    )
    (make block ; rib
        (size t w szwvg)
        (center (/ d 2) 0 zwvg)
        (material SiN)
    )
    (make block ; slab
        (size d infinity szwvg)
        (center (/ t -2) 0 zwvg)
        (material SiN)
    )
))

; source
(define wvlen 1.55)
(define srccomp Ey)
(define zsrc (+ (/ sz -2) spml 0.1))
(set! sources (list (make eigenmode-source
    (src (make gaussian-src
        (wavelength wvlen)
        (width 0.5)
    ))
    (component srccomp)
    (size sx (- sy 2) 0)
    (center 0 0 zsrc)
    (direction Z)
)))

; flux planes (must be done after geometry, sources, resolution, etc.)
(define zfacet (- L (/ sz 2)))
(define fcen (/ 1 wvlen))
(define df 0)
(define nfreq 1)
(define flux11 (add-flux fcen df nfreq
    (make flux-region ; rib
        (size t w 0)
        (center (/ d 2) 0 (+ zsrc 0.1))
    )
    (make flux-region ; slab
        (size d sy 0)
        (center (/ t -2) 0 (+ zsrc 0.1))
    )
))
(define fluxT1 (add-flux fcen df nfreq
    (make flux-region ; rib
```

```
        (size t w 0)
        (center (/ d 2) 0 (+ zfacet 0.1))
    )
    (make flux-region ; slab
        (size d sy 0)
        (center (/ t -2) 0 (+ zfacet 0.1))
    )
))

; ===== RUN THE SIMULATION ==================================================
(use-output-directory (string-append "wvg-out" (number->string h)))
(define decayplus 50)
(define decayval 1e-2)
(define decaypoint (vector3 0 0 (+ zfacet 0.1)))
(if facet? (begin
    (load-minus-flux "flux11-flux" flux11)
))
(run-sources+
    (stop-when-fields-decayed decayplus srccomp decaypoint decayval)
)
(if (not facet?) (begin
    (save-flux "flux11-flux" flux11)
))
(display-fluxes flux11 fluxT1)
```

A.3 Meep installation script

```
# Here is what worked for me to get Meep and MPB working on the
# Tandy supercomputer (http:www.tandysupercomputing.org)
# Phillip Forkner
# March 8, 2016
# phillip.forkner@gmail.com

# the next few lines are probably best put into .bash_profile so that they're
# loaded everytime you login

# load modules
ml gcc blas lapack openmpi libtool libunistring bdwgc guile ffmpeg x264

# check versions of loaded modules
ml
# output for me is:
#    1) gcc/4.4.7-3
#    2) blas/1
#    3) lapack/3.5.0
#    4) openmpi/1.6.5
#    5) libtool/2.4.6
#    6) libunistring/0.9.5
#    7) bdwgc/7.4.2
#    8) guile/2.0.11
#    9) ffmpeg/2.1.4
#    10) x264/2409

# set some environment variables
export PATH=$PATH:/home/pforkner/bin
export LDFLAGS="-L/home/pforkner/lib -L/home/pforkner/share"
export CPPFLAGS=-I/home/pforkner/include
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/pforkner/lib:/home/pforkner/sha

# suppress some guile warnings
export GUILE_WARN_DEPRECATED=no

# Now, install the following from source in $HOME (or whichever path you're
# using):

mkdir downloads
cd downloads

# harminv
wget http://ab-initio.mit.edu/harminv/harminv-1.4.tar.gz
tar -xvzf harminv-1.4.tar.gz
cd harminv-1.4
./configure --prefix=$HOME
make && make install
cd ..

# libctl
wget http://ab-initio.mit.edu/libctl/libctl-3.2.2.tar.gz
```

```
tar -xvzf libctl-3.2.2.tar.gz
cd libctl-3.2.2
./configure --prefix=$HOME
make && make install
cd ..

# hdf5
wget http://www.hdfgroup.org/ftp/HDF5/current/src/hdf5-1.8.16.tar.gz
tar -xvzf hdf5-1.8.16.tar.gz
cd hdf5-1.8.16
./configure --prefix=$HOME --enable-parallel
make && make install
cd ..

# h5utils
wget http://ab-initio.mit.edu/h5utils/h5utils-1.12.1.tar.gz
tar -xvzf h5utils-1.12.1.tar.gz
cd h5utils-1.12.1
./configure --prefix=$HOME
make && make install
cd ..

# fftw
wget http://www.fftw.org/fftw-3.3.4.tar.gz
tar -xvzf fftw-3.3.4.tar.gz
cd fftw-3.3.4
./configure --prefix=$HOME --enable-mpi
make && make install
cd ..

# NOTE BELOW that for some reason I could not use the "$HOME"
# variable in "--with-libctl=" when configuring MPB and Meep.
# So, replace "username" with your username (or whichever path
# you're using) in the three places below.

# MPB (without and with mpi)
wget http://ab-initio.mit.edu/mpb/mpb-1.5.tar.gz
tar -xvzf mpb-1.5.tar.gz
cd mpb-1.5
./configure --prefix=$HOME --with-libctl=home/username/share/libctl
make && make install
make distclean
./configure --prefix=$HOME --with-mpi --with-libctl=home/username/share/libct
make && make install
cd ..

# Meep
wget http://ab-initio.mit.edu/meep/meep-1.3.tar.gz
tar -xvzf meep-1.3.tar.gz
cd meep-1.3
./configure --prefix=$HOME --with-mpi --with-libctl=home/username/share/libct
make && make install
```

```
cd ~
```

A.4 Tandy supercomputer script

```
#!/usr/bin/env bash

#BSUB -q long
#BSUB -n 48
#BSUB -R "span[ptile=16]"
#BSUB -o %J_stdout.txt
#BSUB -e %J_stderr.txt
#BSUB -J "3drib2"

#BSUB -W 10:00

ml gcc blas lapack openmpi libtool libunistring bdwgc guile

for h in {0.35,0.45,0.55,0.6,0.8} ; do
    mkdir wvg-out$h
    mpirun meep-mpi h=$h wvg.ctl >> wvg-out$h/wvg.out
    mpirun meep-mpi h=$h facet?=true wvg.ctl >> wvg-out$h/wvg.out
    printf "$h," >> fluxes.csv
    grep "flux1:" wvg-out$h/wvg.out | tr "\n" "," >> fluxes.csv
    grep "Elapsed" wvg-out$h/wvg.out | tr "\n" "," >> fluxes.csv
    printf "\n" >> fluxes.csv
done
```

A.5 Mathematica™notebook for the effective index method