Introduction to Computational Electromagnetics

Thomas E. Roth

Elmore Family School of Electrical and Computer Engineering
Purdue University

Spring 2024

Preface

This set of lecture notes was prepared for a single-semester advanced-level graduate course on computational electromagnetics offered in the Elmore Family School of Electrical and Computer Engineering at Purdue University. The course assumes that the students have already completed a typical introductory graduate-level course on electromagnetic theory; although, we have frequently taught students from other engineering and science departments that have been able to successfully complete the course despite not having as strong of a background in electromagnetic theory.

These lecture notes focus primarily on the fundamental concepts concerning the three major classes of computational electromagnetics techniques; namely, finite difference methods, finite element methods, and the method of moments. There is a short discussion on fast algorithms as well. We do not touch on many other important topics due to time constraints in a single semester course, but we typically leave the final two weeks of the semester for student-led presentations on applied and advanced topics in computational electromagnetics that usually fill in on many additional topics of interest.

In preparing these lecture notes, we drew heavily upon the fantastic textbook *Theory and Computation of Electromagnetic Fields* by J.-M. Jin. We also treated this textbook as a required reference for the course, with occasional homework assignments taken from the end-of-chapter problems as well. In these lecture notes, we have expanded on the content from this textbook with additional derivation details and references to the literature or other resources to reinforce some of the critical concepts. Although the theory of computational electromagnetics is very important to cover, the “real” learning comes in the form of coding projects that require each student to implement a basic version of each of the three main classes of computational electromagnetics techniques. The projects that are assigned with respect to these methods at Purdue are discussed in the final section of each chapter on a particular computational technique.

Contents

Chapter 1

Introduction and Electromagnetic Theory Review

1.1 Overview of Computational Electromagnetics

The field of *computational electromagnetics (CEM)* involves the study of how to solve Maxwell's equations numerically. Performing this kind of numerical analysis has become an essential part of the design process for many electromagnetic devices, ranging from antennas and microwave components all the way to optical and photonic systems. One of the main reasons for the popularity of these methods in designing real world devices is the amazing predictive power of Maxwell's equations.

Maxwell's equations (completed in 1865) describe the fundamental properties and interplay between electricity and magnetism. Maxwell's equations represent one of the greatest triumphs ever achieved in physics. They are for almost all intents and purposes, perfect. In 2012, their theoretical accuracy had been experimentally validated to one part in a trillion. This level of accuracy is equivalent to measuring the distance from the Earth to the Moon and being correct to within the width of a *single human hair* [1].

Many other disciplines in physics can only dream of having foundational equations that are this accurate or well-behaved. For instance, in the field of fluid dynamics, the Navier-Stokes equations are of practical interest for many applications ranging from modeling weather to complex aerodynamics. Even though they have had great success in practical applications, the proof of existence and smoothness of solutions to the Navier-Stokes equations are still considered one of the most important open problems in mathematics. Such is the importance that the Clay Mathematics Institute has offered a $1M USD bounty for anyone who can solve it!

Further, many areas of physics can only develop equations with various approximations in place – i.e., they are only valid for certain situations and uses, making their application difficult for the many "boundary cases" that inevitably occur in practical engineering analysis and design. In the field of electromagnetics we rarely have to be concerned about this, we can always fall back on the validity of Maxwell's equations to know that we have firm footing to investigate further concepts. We are truly privileged and indebted to the great work of Maxwell and others who were so successful in developing the theory of electromagnetism!

Although Maxwell's equations are essentially perfect, they are in general also extremely difficult to solve except for the simplest of geometries. Recall that Maxwell's equations are given by the following.

Maxwell's Equations

$$\text{Ampere}'\text{s Law}: \quad \nabla \times \mathbf{H} = \partial_t \mathbf{D} + \mathbf{J} \tag{1.1}$$

$$\text{Faraday}'\text{s Law}: \quad \nabla \times \mathbf{E} = -\partial_t \mathbf{B} - \mathbf{M} \tag{1.2}$$

$$\text{Gauss}'\text{ Law of Electricity}: \quad \nabla \cdot \mathbf{D} = \rho \tag{1.3}$$

$$\text{Gauss}'\text{ Law of Magnetism}: \quad \nabla \cdot \mathbf{B} = \rho_m \tag{1.4}$$

These equations relate the following electromagnetic quantities to each other.

- $\mathbf{E}(\mathbf{r}, t)$: electric field intensity [V/m]
- $\mathbf{H}(\mathbf{r}, t)$: magnetic field intensity [A/m]
- $\mathbf{D}(\mathbf{r}, t)$: electric flux density [C/m^2]
- $\mathbf{B}(\mathbf{r}, t)$: magnetic flux density [T = Wb/m^2]

In addition to electric and magnetic fields/fluxes, there are also four additional sources in Maxwell's equations that can produce electric and magnetic fields.

- $\mathbf{J}$: electric current density [A/m^{0,1,2}]
- ρ: electric charge density [C/m^3]
- $\mathbf{M}$: magnetic current density [V/m^{0,1,2}]
- ρ_m: magnetic charge density [Wb/m^3]

The electric sources are related by the current continuity equation, which is

$$\nabla \cdot \mathbf{J} = -\partial_t \rho. \tag{1.5}$$

A similar equation also holds for the magnetic sources. However, it is important to remember that the magnetic sources are generally added to Maxwell's equations as a mathematical mechanism to assist in finding the solution to a problem. In reality, magnetic currents and charges do not exist.

Before moving on, it is good to take a closer look at the units of **E**, **H**, **D**, and **B**. From this, we see that the electric and magnetic *fields* each have a unit of 1/m, while the electric and magnetic *fluxes* each have a unit of 1/m^2. As we become more and more acquainted with these quantities, we can often have a tendency to treat the fields and fluxes as interchangeable. However, it is always good to remember that there are some fundamental (and important) differences, as suggested by the units. In particular, we should always think of **E** and **H** as quantities that are "built" to be integrated along a line/curve, while **D** and **B** are "built" to be integrated along surfaces.

Even very talented researchers have fallen into the trap of treating the fields and fluxes as interchangeable time and time again! There are many instances throughout the history of CEM where the researchers forgot this core principle. These early numerical methods would produce "spurious" or erroneous solutions that were sometimes difficult to properly diagnose. However, in many cases, the issue was that **E** and **H** weren't being treated as quantities "built" to be integrated along a curve, while **D** and **B** weren't being treated as quantities "built" to be integrated over a surface. Once these quantities were treated appropriately, all these spurious numerical issues vanished!

A quick inspection of Maxwell's equations shows us that we need additional equations to have a solvable system (we have 8 scalar equations and 12 scalar unknowns). We can relate the fields and fluxes to one another in terms of the constitutive relations. For simple materials, **D** can be related to **E**, while **B** can be related to **H**. The particular relationships are

$$\mathbf{D} = \epsilon \mathbf{E}, \tag{1.6}$$

$$\mathbf{B} = \mu \mathbf{H}, \tag{1.7}$$

where ϵ is the *permittivity* of the material (with units of [F/m]) and μ is the *permeability* of the material (with units [H/m]). Recall our discussion on units and the distinction in character between the fields and fluxes (in terms of what they should be integrated over). We see that even though these constitutive relationships appear very simple, they are fundamentally changing the character of the fields into fluxes (and, in some sense, vice-versa). Looking at the units we also see that ϵ will augment/contribute to capacitive effects, while μ will do the same for inductive effects.

Beyond this, it is important to remember that strictly speaking these constitutive relations are only valid in this simple form for a non-dispersive medium (i.e., they do not vary as a function of frequency). If we perform a frequency domain analysis (more on this later), these expressions will hold as a function of frequency. However, in the time domain, these expressions should be more generally written as a convolution between the constitutive parameters and the relevant fields.

Another kind of constitutive relationship that exists is Ohm's law. In the full electromagnetic picture, Ohm's law is

$$\mathbf{J} = \sigma \mathbf{E}, \tag{1.8}$$

where σ is the *conductivity* of the material. Typically, a unit of [S/m] is used for the conductivity. Again note how the units taking part in this constitutive relation is fundamentally changing the character of the electric field (i.e., the current density on the left should be integrated over a surface).

Considering all these points, we see that Maxwell's equations correspond to a complex system of coupled partial differential equations (PDEs). It is only for relatively simple situations that we can compute a complete analytical solution to a problem. In general, the few cases that we can solve analytically require the geometry we are considering to have some kind of symmetry that can be exploited within a simple coordinate system (e.g., circular/cylindrical or spherical symmetry). Advanced analytical techniques can be applied to develop good approximate solutions for other situations.

However, the art of developing these kinds of solutions is becoming less and less common due to the success and power of CEM techniques. These newer methods can be applied to an extremely wide range of problems using the same underlying technique/code. Due to this adaptability, they can produce highly accurate results that agree extremely well with measurements for a wide variety of practical applications.

Although many different CEM techniques have been developed over the years, in this course we will predominantly focus on the three main methods that have been developed: the finite-difference time-domain (FDTD) method, the finite element method (FEM), and the method of moments (MoM). Many other methods can be viewed as subsets of these general techniques, and can be quickly learned once you have grasped the fundamentals of these three methods. Before diving into a discussion of these different CEM techniques, we will first review some essential aspects of electromagnetic theory that will be useful in learning CEM.

1.2 Electromagnetic Theory Review

1.2.1 Integral Form of Maxwell's Equations

We will begin by reviewing how Maxwell's equations can be converted into their integral forms. Let's start with Ampere's law. From our previous discussions, we see that all the quantities in this equation are "built" to be integrated over a surface.

Considering this, we can integrate Ampere's law over surface S to give us

$$\iint_S \big(\nabla \times \mathbf{H}\big) \cdot \hat{n}\, dS = \iint_S \big(\partial_t \mathbf{D} + \mathbf{J}\big) \cdot \hat{n}\, dS. \tag{1.9}$$

Looking at the left-hand side, we see that we can simplify this by using Stokes' theorem. The result is

$$\oint_C \mathbf{H} \cdot d\boldsymbol{\ell} = \iint_S \big(\partial_t \mathbf{D} + \mathbf{J}\big) \cdot \hat{n}\, dS. \tag{1.10}$$

We see that the magnetic field circulating a surface depends on the total current flowing through that surface (where $\partial_t \mathbf{D}$ is typically referred to as the *displacement current*). Next, let's consider Faraday's law. The same treatment with Stokes' theorem can be applied here to give us

$$\oint_C \mathbf{E} \cdot d\boldsymbol{\ell} = -\iint_S \big(\partial_t \mathbf{B} + \mathbf{M}\big) \cdot \hat{n}\, dS. \tag{1.11}$$

Now let's turn our attention to Gauss' laws of electricity and magnetism. In both cases, we are taking the divergence of the fluxes. Hence, these equations are giving us quantities that should be integrated over a volume. Doing this gives us

$$\iiint \nabla \cdot \mathbf{D}\, dV = \iiint \rho\, dV \tag{1.12}$$

for Gauss' law of electricity. Looking at the left-hand side, we see that we can simplify this using the divergence theorem. This gives

$$\oiint_S \mathbf{D} \cdot \hat{n}\, dS = \iiint \rho\, dV. \tag{1.13}$$

In words, we see that the total flux exiting some volume depends on the total amount of charge contained within that volume. Following an identical process for Gauss' law of magnetism gives us

$$\oiint_S \mathbf{B} \cdot \hat{n}\, dS = \iiint \rho_m\, dV. \tag{1.14}$$

In the usual case of no magnetic charges ($\rho_m = 0$), this equation tells us two (related) things. First, because there are no magnetic charges the flux leaving a closed surface will always be exactly balanced out by an equal amount of flux entering the surface. Stated another way, this tells us that $\mathbf{B}$ always forms closed loops.

In summary, the integral form of Maxwell's equations are the following.

Maxwell's Equations (Integral Form)

$$\text{Ampere's Law}: \quad \oint_C \mathbf{H} \cdot d\boldsymbol{\ell} = \iint_S \big(\partial_t \mathbf{D} + \mathbf{J}\big) \cdot \hat{n}\, dS \tag{1.15}$$

$$\text{Faraday's Law}: \quad \oint_C \mathbf{E} \cdot d\boldsymbol{\ell} = -\iint_S \big(\partial_t \mathbf{B} + \mathbf{M}\big) \cdot \hat{n}\, dS \tag{1.16}$$

$$\text{Gauss' Law of Electricity}: \quad \oiint_S \mathbf{D} \cdot \hat{n}\, dS = \iiint \rho\, dV \tag{1.17}$$

$$\text{Gauss' Law of Magnetism}: \quad \oiint_S \mathbf{B} \cdot \hat{n}\, dS = \iiint \rho_m\, dV \tag{1.18}$$

1.2.2 Boundary Conditions

You should be able to recall that the differential forms of Maxwell's equations are incomplete without also knowing the boundary conditions for the different fields and fluxes. These can be readily derived from the integral form of Maxwell's equations. We will not review this derivation here. If you need a refresher on how this derivation is completed, it can be found in most textbooks on electromagnetic theory. The main important results are the following.

Boundary Conditions

$$\hat{n} \times (\mathbf{E}_1 - \mathbf{E}_2) = -\mathbf{M}_s \tag{1.19}$$

$$\hat{n} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s \tag{1.20}$$

$$\hat{n} \cdot (\mathbf{D}_1 - \mathbf{D}_2) = \rho_s \tag{1.21}$$

$$\hat{n} \cdot (\mathbf{B}_1 - \mathbf{B}_2) = \rho_{m,s} \tag{1.22}$$

The sign convention for these boundary conditions are established for $\hat{n}$ being the unit normal vector pointing from Region 2 into Region 1. Further, the subscripts s on the various right-hand sides indicate that these are surface current and charge densities. In most traditional cases, these surface densities only exist at the interface between a dielectric region and a perfect conductor (either electric or magnetic) or will be specified as a part of a problem statement.

More broadly, boundary conditions are often classified into three main groups when studying PDEs. First, there are *Dirichlet boundary conditions* that correspond to specifying the value of the function being solved on the boundary surfaces. Explicitly, if we are solving for f in a region Ω then a Dirichlet boundary condition is typically of the form

$$f(\mathbf{r}) = g(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega, \tag{1.23}$$

where g is some given function. If $g(\mathbf{r}) = 0$, then this is referred to as a *homogeneous Dirichlet boundary condition.* This kind of boundary condition is also sometimes called a *boundary condition of the first kind.*

Another common type of boundary condition is a *Neumann boundary condition* (or *boundary condition of the second kind*). These boundary conditions place a constraint on the normal derivative of the quantity being solved for. For instance, if we are solving for f in a region Ω then a Neumann boundary condition could be specified as

$$\partial_n f(\mathbf{r}) = g(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega, \tag{1.24}$$

where g is some given function. If $g(\mathbf{r}) = 0$, then this is referred to as a *homogeneous Neumann boundary condition.*

The final main kind of boundary condition is a *Robin boundary condition.* These are also referred to as *boundary conditions of the third kind.* These place a constraint on the linear combination of the function and its normal derivative. For instance, we can have

$$f(\mathbf{r}) + h(\mathbf{r})\partial_n f(\mathbf{r}) = g(\mathbf{r}), \quad \mathbf{r} \in \partial\Omega, \tag{1.25}$$

where g and h are given functions. Again, if $g(\mathbf{r}) = 0$ then this would be referred to as a *homogeneous Robin boundary condition.*

From our earlier electromagnetic boundary conditions, we can see that these are nominally all written as Dirichlet boundary conditions. However, when we solve the wave equation only in terms of $\mathbf{E}$ or $\mathbf{H}$ then we can have Neumann or Robin boundary conditions appearing in certain situations. We will encounter all of these conditions throughout this class when we consider solving problems with boundary conditions corresponding to perfect electric conductors, perfect magnetic conductors, and impedance surfaces.

1.2.3 Frequency Domain Representation

In general, solving Maxwell's equations in the time domain tends to be a difficult undertaking even when using computational methods. We will study time domain methods in this class, however, it will also be advantageous to develop frequency domain methods. To begin, let's recall some basic details about the Fourier transform.

Fourier Transform

The Fourier transform provides us with a way to express a time domain signal in terms of an "infinite summation" of sinusoidal functions (and vice-versa). If you recall your linear algebra theory, the Fourier transform can be predominantly viewed as a kind of basis transformation. The particular form of the Fourier transform that we will use in this course is

$$f(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt. \tag{1.26}$$

We can interpret this integration as an inner product of our function $f(t)$ with a particular *Fourier harmonic*, namely, $e^{j\omega t}$. Hence, we are measuring how much of $f(t)$ "overlaps" with the particular Fourier harmonic of interest.

We can invert our Fourier transform using a similar transform. In particular, this will be

$$f(t) = \mathcal{F}^{-1}\{f(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(\omega) e^{j\omega t} d\omega, \tag{1.27}$$

which can be interpreted as an "infinite summation" of all the different Fourier harmonics.

One of the most important properties of the Fourier transform for simplifying differential equations is how a derivative of a Fourier transform variable changes. For the Fourier transform convention we will use in this class, we see that

$$\boxed{\frac{d}{dt} \Longleftrightarrow j\omega} \tag{1.28}$$

where the $\Longleftrightarrow$ denotes how these quantities transform under the action of the Fourier and inverse Fourier transforms. It should be stressed that the sign for this identity can be different if the opposite sign convention is used for the Fourier transform. *This happens frequently in the physics and CEM literature, so it is best to be careful and always be cognizant of what sign convention is being used when comparing results from different sources.*

Let's see how we can use this to simplify Maxwell's equations. We will start by assuming that we are dealing with a linear, time-invariant system so that our Fourier representations of functions are appropriate. Now, let's rewrite all of the quantities in Ampere's law using the representation provided by the inverse Fourier transform. This gives us

$$\begin{aligned}\nabla \times \mathbf{H}(\mathbf{r},t) &= \partial_t \mathbf{D}(\mathbf{r},t) + \mathbf{J}(\mathbf{r},t)\\ \nabla \times \left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{H}(\mathbf{r},\omega)e^{j\omega t}d\omega\right] &= \partial_t\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{D}(\mathbf{r},\omega)e^{j\omega t}d\omega\right] + \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{J}(\mathbf{r},\omega)e^{j\omega t}d\omega\\ \frac{1}{2\pi}\int_{-\infty}^{\infty} \nabla \times \mathbf{H}(\mathbf{r},\omega)e^{j\omega t}d\omega &= \frac{1}{2\pi}\int_{-\infty}^{\infty} j\omega\mathbf{D}(\mathbf{r},\omega)e^{j\omega t}d\omega + \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathbf{J}(\mathbf{r},\omega)e^{j\omega t}d\omega\end{aligned} \tag{1.29}$$

Since our system is linear and time-invariant, we can analyze this system at each frequency component individually and then add up all the results later to recover the full time domain response. Due to this, we don't need to "worry" about the integrations and common terms in this equation. A more exact justification for this is possible by "projecting" our equation onto a particular Fourier harmonic of interest. We can do this by taking the inner product of (1.29) with a Fourier harmonic, e.g., $\exp[j\omega' t]$. For notational simplicity, we will only focus on what happens to the left-hand side of this equation (the same process will work for all of the terms in the overall equation). Now, taking the inner product gives us

$$\int_{-\infty}^{\infty} e^{-j\omega' t}\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \nabla \times \mathbf{H}(\mathbf{r},\omega)e^{j\omega t}d\omega\right]dt. \tag{1.30}$$

Swapping the order of integrations then gives us

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} \nabla \times \mathbf{H}(\mathbf{r},\omega)\left[\int_{-\infty}^{\infty} e^{j(\omega-\omega')t}dt\right]d\omega. \tag{1.31}$$

The inner integration will only be non-zero if $\omega = \omega'$. More generally, we can recognize this inner integration as being proportional to the Dirac delta function $\delta(\omega-\omega')$. Hence, we have that

$$\int_{-\infty}^{\infty} \nabla \times \mathbf{H}(\mathbf{r},\omega)\delta(\omega-\omega')d\omega. \tag{1.32}$$

This final integration can be evaluated easily using the sifting property of the delta function. This finally gives us the simplified result that

$$\int_{-\infty}^{\infty} e^{-j\omega' t}\left[\frac{1}{2\pi}\int_{-\infty}^{\infty} \nabla \times \mathbf{H}(\mathbf{r},\omega)e^{j\omega t}d\omega\right]dt = \nabla \times \mathbf{H}(\mathbf{r},\omega'). \tag{1.33}$$

This same process can be applied to the rest of the terms in Ampere's law given in (1.29). By finally swapping ω' to ω at the end to keep with a consistent notation, we have

$$\nabla \times \mathbf{H}(\mathbf{r},\omega) = j\omega\mathbf{D}(\mathbf{r},\omega) + \mathbf{J}(\mathbf{r},\omega). \tag{1.34}$$

We can follow suit with the rest of Maxwell's equations. The end result is summarized below.

Maxwell's Equations (Frequency Domain)

$$\text{Ampere's Law}: \quad \nabla \times \mathbf{H} = j\omega\mathbf{D} + \mathbf{J} \tag{1.35}$$

$$\text{Faraday's Law}: \quad \nabla \times \mathbf{E} = -j\omega\mathbf{B} - \mathbf{M} \tag{1.36}$$

$$\text{Gauss' Law of Electricity}: \quad \nabla \cdot \mathbf{D} = \rho \tag{1.37}$$

$$\text{Gauss' Law of Magnetism}: \quad \nabla \cdot \mathbf{B} = \rho_m \tag{1.38}$$

Similarly, the current continuity equation becomes

$$\nabla \cdot \mathbf{J} = -j\omega\rho. \tag{1.39}$$

In electrical engineering, we typically use *phasors* to represent our time-harmonic quantities. These are completely specified if we establish their amplitude, frequency, and initial phase. We do this so that we can represent our phasors using exponentials (for which algebra is very easy) as opposed to the more cumbersome trigonometric functions. For a cosine reference, an example of this is

$$\begin{aligned} E_x(z,t) &= E_0 \cos\big(\omega t - \varphi\big) \\ &= \mathrm{Re}\Big\{ \underline{E_0 e^{-j\varphi}}\, e^{j\omega t} \Big\}. \end{aligned} \tag{1.40}$$

We typically call the underlined portion the phasor. It has an amplitude of E_0, a phase of φ, and a frequency of ω. We can see that these equations are equivalent by using Euler's formula to write the exponential functions as trigonometric functions. We will use this phasor representation in this course. Although we will not frequently need to use it, it is good to remember that we can recover a time domain representation of the behavior of a phasor solution by multiplying it by $\exp(j\omega t)$ and taking the real part. This can be useful for generating time-harmonic visualizations of some simulations produced by various frequency domain CEM techniques.

1.2.4 Uniqueness Theorem

Although many electromagnetic theorems can be useful in developing CEM methods or applying them to studying particular problems, the uniqueness theorem is of particular importance. This is because the uniqueness theorem helps us to know what conditions we need to specify to have a *well-posed* mathematical description of a physical system of interest. This is essential in making sure that we have enough information built into our physical model before we try and apply a numerical method to solve it.

To formulate the uniqueness theorem, we originally assume that for the same problem (specified in terms of some configuration of $\mathbf{M}$, $\mathbf{J}$, μ, ϵ, and σ) we have two different electric and magnetic fields produced within a volume V. If we subtract Maxwell's equations from each other for these two cases we end up with:

$$\nabla \times \boldsymbol{\delta E} = -j\omega\mu\boldsymbol{\delta H}, \tag{1.41}$$

$$\nabla \times \boldsymbol{\delta H} = j\omega\epsilon\boldsymbol{\delta E} + \sigma\boldsymbol{\delta E}, \tag{1.42}$$

where $\boldsymbol{\delta E}$ and $\boldsymbol{\delta H}$ are the differences in the electric and magnetic fields produced for the same problem setup, respectively. We can combine these equations in a form reminiscent of Poynting's theorem to arrive at power quantities by taking

$$\begin{aligned}\boldsymbol{\delta H}^* \cdot \nabla \times \boldsymbol{\delta E} - \boldsymbol{\delta E} \cdot \nabla \times \boldsymbol{\delta H}^* &= \nabla \cdot (\boldsymbol{\delta E} \times \boldsymbol{\delta H}^*) \\ &= -j\omega\mu|\boldsymbol{\delta H}|^2 + (j\omega\epsilon^* - \sigma)|\boldsymbol{\delta E}|^2.\end{aligned} \tag{1.43}$$

We can integrate this over the volume of interest and apply the divergence theorem to get

$$\oiint_S (\boldsymbol{\delta E} \times \boldsymbol{\delta H}^*) \cdot \hat{n} dS = \iiint_V \big[- j\omega\mu|\boldsymbol{\delta H}|^2 + (j\omega\epsilon^* - \sigma)|\boldsymbol{\delta E}|^2\big] dV. \tag{1.44}$$

For us to "prove" that the two electromagnetic fields must be identical for the same situation (i.e., unique), we want to consider some possible ways for this equation to be satisfied. The traditional argument is that if some situation exists that causes the left-hand side to be zero, then we can guarantee that $\boldsymbol{\delta E} = \boldsymbol{\delta H} = 0$ throughout all of V if the medium in V is lossy and the frequency is non-zero. We can then conceive of the static and lossless cases as limiting situations of the time-varying and lossy cases to establish the uniqueness of the electromagnetic fields.

Now, the main ways of interest to ensure that the left-hand side of (1.44) equals zero are:

1. the tangential field ($\hat{n} \times \mathbf{E}$) is specified over the entire surface of S so that $\hat{n} \times \boldsymbol{\delta E} = 0$ on S,

2. the tangential field ($\hat{n} \times \mathbf{H}$) is specified over the entire surface of S so that $\hat{n} \times \boldsymbol{\delta H} = 0$ on S,

3. or some combination of the tangential electric and magnetic fields are specified over the entire surface of S.

The important result of this is that for us to develop a well-posed mathematical formulation of a problem we will need to specify what value the tangential electric or magnetic fields take on the boundaries of the region we are analyzing numerically. Without having this boundary data specified, we typically will be unable to solve a problem.

Chapter 2

Finite Difference Methods

The first CEM method we will learn about in this course is the *finite difference method*, with a particular focus on the *finite-difference time-domain (FDTD) method.* The basic form of the FDTD method is one of the simplest CEM methods to understand and implement, which has led to it becoming an extremely popular technique in many research communities.

The basic process of any finite differencing method is to first take a continuous problem and discretize it into some form of structured grid. The differential equation to be solved is then reduced into a *difference equation* that only involves quantities that are available at the discrete points of the structured grid. The process of converting a differential equation into a differencing equation can be performed using a number of standard *finite differencing formulas*, which we will discuss in detail shortly. The final step of a finite differencing method is to then solve the difference equations by using some kind of time- and/or space-stepping scheme. This involves finding a way to rearrange the various differencing equations in such a way that there is only a single unknown quantity in each equation that can be computed easily from the previously computed quantities.

2.1 Finite Differencing Formulas

As alluded to earlier, the heart of a finite differencing method is the particular finite differencing formula that is used to convert the differential equation into a difference equation. There are three basic options for converting a derivative to a finite difference form. The basic idea is to make an approximation to the definition of a derivative. You should be able to recall that by definition a derivative can be computed as

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}. \tag{2.1}$$

A finite difference approximation to this formula is to just stop the limit for some particular value of Δx. If this Δx is small enough compared to the underlying variation of $f(x)$, then we can treat

$$f'(x) \approx \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{2.2}$$

as a reasonable approximation to a derivative for some non-zero Δx. This particular formula is known as a *forward difference* because it requires us to be able to evaluate the function $f(x)$ at a location forward along our discrete grid then where we are evaluating the derivative at.

This is of course not the only option. We can easily see that another valid option would be

$$f'(x) \approx \frac{f(x) - f(x - \Delta x)}{\Delta x}. \tag{2.3}$$

This is known as a *backward difference*, which has similar properties to the forward difference. However, depending on the equations being discretized and the information that is known, the backward difference formula may be more appropriate in certain contexts.

The final main option that is used in finite differencing methods is known as the *central difference*. The formula for a central difference can be easily established by adding (2.2) and (2.3) together. This gives

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}. \tag{2.4}$$

This formula is often preferred over the forward or backward differencing formulas if the application allows.

Why is this the case? The answer lies in the *accuracy* of the approximation that each formula achieves to the derivative attempting to be computed. The easiest way to establish the *order of the approximation* is to use the Taylor series representation of the function. For instance, we can readily find that

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)(\Delta x)^2 + \frac{1}{6}f'''(x)(\Delta x)^3 + \dots, \tag{2.5}$$

which can be rearranged as

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x). \tag{2.6}$$

This shows that the forward differencing formula is only *first-order accurate* because the leading error term is proportional to $(\Delta x)^p$ with $p = 1$ (had the leading error term had $p = 2$ the formula would be *second-order accurate*). Generally, a first-order accurate method is considered to be rather poor for most computational methods (although exceptions to this certainly exist for particularly difficult problems to solve).

A similar analysis can be performed for the backward differencing formula. For this case, the Taylor series is

$$f(x - \Delta x) = f(x) - f'(x)\Delta x + \frac{1}{2}f''(x)(\Delta x)^2 - \frac{1}{6}f'''(x)(\Delta x)^3 + \dots, \tag{2.7}$$

which can be rearranged as

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + O(\Delta x). \tag{2.8}$$

It should not be too surprising that this also achieves first-order accuracy given the similar structure to the forward differencing formula.

Now, something interesting happens when we derive the central differencing formula from the Taylor series representation. To do this, we can subtract (2.7) from (2.5) to get

$$f(x+\Delta x) - f(x-\Delta x) = 2f'(x)\Delta x + \frac{1}{3}f'''(x)(\Delta x)^3 + \dots. \tag{2.9}$$

From this, we see that the leading order error terms from the previous examples have canceled. As a result, when we solve this for $f'(x)$ we get that

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O\big((\Delta x)^2\big), \tag{2.10}$$

which we can recognize as being *second-order accurate*. As a result, a coarser discretization size can be used with a central differencing method to achieve the same level of precision as either the forward or backward differencing formulas. This savings in discretization size can be very valuable when attempting to solve large, complicated problems. Hence, central differencing formulas are favored if a suitable method can be developed using them.

In CEM, it is not uncommon to have second-order derivatives in the equations attempting to be solved (e.g., the wave equation). Establishing finite differencing formulas for second-order derivatives can be achieved in a number of ways. In principle, one can derive an approximation to a second-order derivative by applying any of the finite difference approximations for first-order derivatives twice. For example, we could apply forward differencing twice to get

$$f''(x) \approx \frac{f'(x+\Delta x) - f'(x)}{\Delta x} = \frac{f(x+2\Delta x) - 2f(x+\Delta x) + f(x)}{(\Delta x)^2}. \tag{2.11}$$

However, we could also have applied any other differencing formula, making the multiplicity of options rather large. In general, many of these options are not of particular interest and so are rarely derived. Instead, it is most common to use the central differencing formula twice to get

$$f''(x) \approx \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2}. \tag{2.12}$$

This formula can also be derived from the Taylor series approach, which shows that it is second-order accurate in a similar manner to the second-order accuracy of the central difference approximation of a first-order derivative.

2.2 Solution of 1D Electromagnetic Equations

In this section, we will consider an example of how the finite differencing approximations we developed in the previous section can be used to derive a space- and time-marching scheme to solve a 1D electromagnetic equation. In particular, we will consider the analysis of a lossless transmission line geometry using the telegrapher's equations. Recall that the telegrapher's equations for this case are

$$\partial_x V = -L\partial_t I, \tag{2.13}$$

$$\partial_x I = -C\partial_t V - I_S, \tag{2.14}$$

where V and I are the voltage and current on the line, I_S is a current source, and L and C are the per-unit-length inductance and capacitance, respectively. For an arbitrary transmission line, L and C can change values as a function of position. The underlying variation of these parameters is sometimes omitted to simplify the notation, with the understanding that when coding the equations the correct values of L and C are used as needed (will consider how to handle this correctly in a more general context later in the course). We will now consider two ways to go about solving the telegrapher's equations using a finite difference method.

2.2.1 Time-Marching a Second-Order Equation

The first approach we will consider involves solving a wave equation. To begin, we first combine the telegrapher's equations into a wave equation for V or I. Considering the case for V, we have

$$\partial_x^2 V - LC\partial_t^2 V = L\partial_t I_S. \tag{2.15}$$

We can now follow the process for developing a finite differencing method for this equation. The first step is to discretize the system onto a discrete spatial and temporal grid. In particular, we will only consider the values of quantities at the grid points defined by

$$x = i\Delta x, \quad i = 0, 1, 2, \dots M, \tag{2.16}$$

$$t = n\Delta t, \quad n = 0, 1, 2, \dots N. \tag{2.17}$$

To keep the differencing equations to a manageable size, it is useful to introduce the shorthand notation that

$$V^n(i) = V(i\Delta x, n\Delta t), \tag{2.18}$$

where a similar notation can be used for the other quantities in (2.15) as well.

Our next step in developing our first finite differencing method is to use finite differencing approximations to write (2.15) as a difference equation. If we utilize central differencing approximations for all of the derivatives, we will have that

$$\frac{V^n(i+1) - 2V^n(i) + V^n(i-1)}{(\Delta x)^2} - LC\frac{V^{n+1}(i) - 2V^n(i) + V^{n-1}(i)}{(\Delta t)^2} = L\frac{I_S^{n+1}(i) - I_S^{n-1}(i)}{2\Delta t}. \tag{2.19}$$

We can rearrange this into a *time-stepping formula* by solving for $V^{n+1}(i)$. Doing this, we arrive at

$$V^{n+1}(i) = 2V^n(i) - V^{n-1}(i) + \frac{(\Delta t)^2}{LC(\Delta x)^2}\big[V^n(i+1) - 2V^n(i) + V^n(i-1)\big] \\ - \frac{\Delta t}{2C}\big[I_S^{n+1}(i) - I_S^{n-1}(i)\big]. \quad (2.20)$$

So long as we have all the required information, we can use this formula to compute $V^{n+1}(i)$ at all values of i. Once we have done this, we can use this information to advance to the next time step of the simulation and reuse this time-stepping formula to continue to march forward in time through the simulation. This kind of approach is often referred to as a *time marching* method.

For particularly complex equations, it can be useful to draw a dependency diagram for a particular difference equation to determine whether it will be possible to use it in a time marching method. An example of such a diagram for the simple time stepping formula given in (2.20) is shown in Fig. 2.1. This is a good time-stepping formula because all the needed data comes from previously computed quantities. This kind of time-stepping formula is referred to as an *explicit method* because it only uses known quantities. If there were interdependencies on values that need to be computed concurrently, the time-stepping formula would require the solution of a linear system of equations (e.g., a matrix equation). This kind of time-stepping formula is referred to as an *implicit method* because it requires the solution of a linear system of equations in every time step of the method. An implicit time-marching method is not uncommon for other, more "advanced" CEM techniques. However, this is not as typical for a finite differencing method, but does occur for certain specialized applications.

One important point about the dependency diagram is what happens at the edges or *boundaries* of the diagram. This is where the initial conditions or boundary conditions of the problem come into play. If these are not specified completely and correctly, it is easy for a finite differencing method to produce unwanted/erratic behavior. Hence, it is important to carefully consider the appropriate set of initial conditions and boundary conditions to have a solid mathematical description of a particular problem. For instance, for the problem at hand, we will need to have initial conditions for two time steps worth of data, e.g., for $V^{-2}(i)$ and $V^{-1}(i)$. This is linked to the fact that we have a second-order time derivative in our equation.

Even if we have the initial condition data given to us, it can be important to make sure that the data doesn't unintentionally cause any numerical difficulties. For instance, for most practical purposes we often think of sine and cosine functions as being more or less interchangeable (they only differ by a small phase shift, after all). However, if not implemented carefully, using a cosine function as an initial condition can lead to poorer numerical results compared to if a sine function was used. The underlying issue is that the cosine function would switch from having no values (the source is off) to suddenly having a very large value (the max the source has to offer). This quick change in value can unintentionally excite frequency content in the simulation much higher than intended, leading to unpredictable behavior. In contrast to this, having an initial condition following a sine function will lead to a much gentler transition to a source being turned "on", and hence

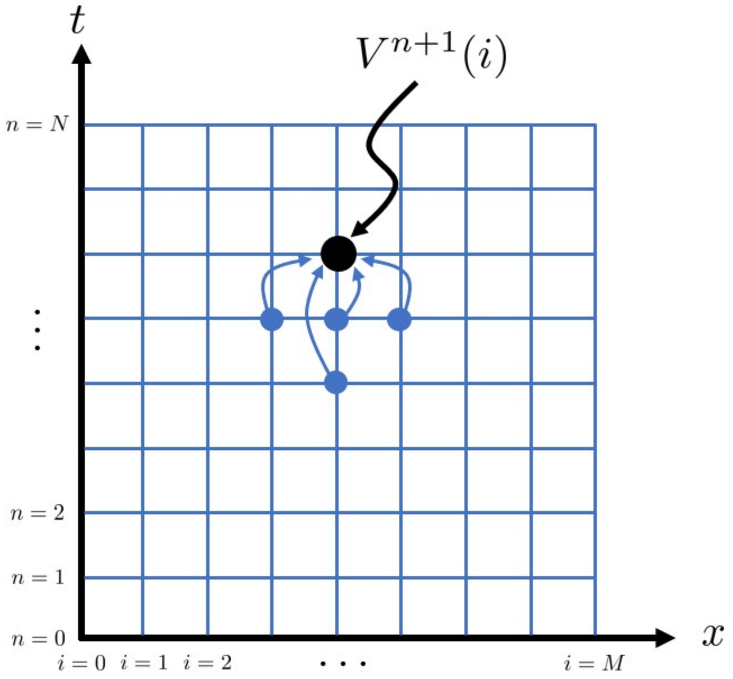


Figure 2.1: Example of a dependency diagram for (2.20). All values that are depended on can be found from previously computed values, making this a suitable explicit time-stepping formula.

can have better numerical performance. In reality, there are many ways to properly handle "gently" exciting a time domain simulation to avoid these kinds of spurious numerical issues. However, it is a good reminder of the intricacy and care that can be needed in developing a numerical algorithm. Seemingly innocuous changes can have substantial impacts, so it is essential to take a methodical and careful approach to developing numerical solvers.

Other data that we need to be able to use the time-stepping formula of (2.20) comes from the boundary conditions. These are used to specify what value the simulation results should take at the edges of the simulation/computational domain. In general, we need some way to terminate the problem we are considering. For a transmission line problem, we may have a short or open circuit termination at either end of the simulation region. For a short circuit, we know that

$$V^n(i) = 0, \text{ if } x = i\Delta x \text{ is a short circuit termination.} \tag{2.21}$$

This kind of boundary condition is known as a *homogeneous Dirichlet condition*. Here, homogeneous refers to the fact that the boundary data is equal to 0 rather than some other functional value and Dirichlet refers to the fact that this specifies the boundary data for the quantity we are solving for. Another kind of boundary condition that is commonly encountered in CEM is known as a *Neumann condition*. This specifies the value of the normal derivative of a function at the boundary of the computational domain. For a transmission line problem, the "normal" aspect is irrelevant, and so it reduces to just specifying the spatial derivative of V at some point. This would be relevant for an open circuit, e.g., because the

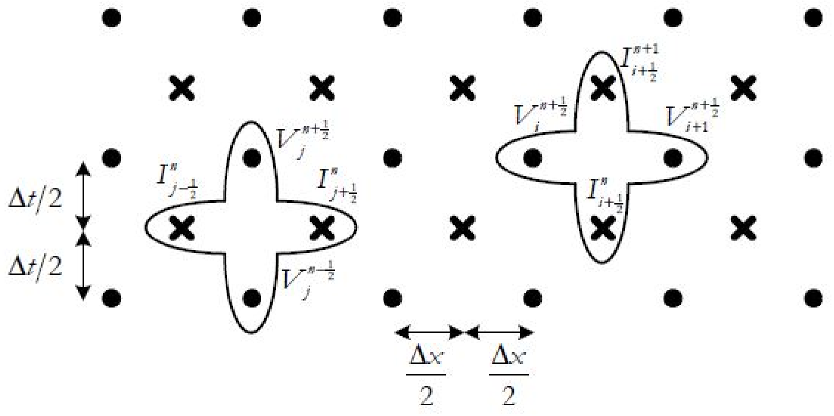


Figure 2.2: Example of a dependency diagram for a leap-frog method for solving the telegrapher's equations (image from [2]).

spatial derivative of V is related to the current (which must vanish at the open circuit) through the telegrapher's equations.

Other more complicated kinds of boundary conditions can also be used for certain problems. These can include Robin boundary conditions that involve both Dirichlet and Neumann conditions. We will also see that terminating an "open" problem (that extends in principle to infinitely far away from the geometry for situations like antenna analysis) requires special care in solving a differential equation. Some of these considerations simplify greatly for 1D analysis, so we will not dwell on them in detail here. Regardless of the particular situation, it is necessary to be careful in implementing a boundary condition within a finite difference method. It is easy to develop a second-order accurate differencing equation that then gets corrupted by a boundary condition that is only implemented to first-order accuracy. Similarly, a poorly implemented boundary condition has the possibility of causing a simulation to become unstable (we will discuss stability in more detail shortly). Hence, boundary conditions should be considered carefully and not be treated as an afterthought in the development of a finite difference method!

2.2.2 Leap-Frog Time-Marching Coupled First Order Equations

The second way to solve the telegrapher's equations is to discretize them both directly. This leads to a set of coupled first order differential equations that must be solved together. However, we cannot simply apply our finite differencing formulas to these equations directly in a naive manner or we will end up with a set of equations for which we can't develop an appropriate explicit time-marching method. The clever way around this problem is to discretize the voltage and current on staggered grids in both space and time, as shown in Fig. 2.2. By adopting these staggered grids, an explicit time-marching method can be developed that uses central differences for all derivatives, and is thus second-order accurate.

If we ignore the presence of sources, we can discretize (2.13) and (2.14) as

$$\frac{I^n(i+\frac{1}{2}) - I^n(i-\frac{1}{2})}{\Delta x} = -C\frac{V^{n+\frac{1}{2}}(i) - V^{n-\frac{1}{2}}(i)}{\Delta t}. \tag{2.22}$$

$$\frac{V^{n+\frac{1}{2}}(i+1)-V^{n+\frac{1}{2}}(i)}{\Delta x}=-L\frac{I^{n+1}(i+\frac{1}{2})-I^{n}(i+\frac{1}{2})}{\Delta t}, \tag{2.23}$$

We can rearrange both of these into time-stepping formulas to get

$$V^{n+\frac{1}{2}}(i)=V^{n-\frac{1}{2}}(i)-\frac{\Delta t}{C\Delta x}\left[I^{n}\left(i+\frac{1}{2}\right)-I^{n}\left(i-\frac{1}{2}\right)\right], \tag{2.24}$$

$$I^{n+1}\left(i+\frac{1}{2}\right)=I^{n}\left(i+\frac{1}{2}\right)-\frac{\Delta t}{L\Delta x}\left[V^{n+\frac{1}{2}}(i+1)-V^{n+\frac{1}{2}}(i)\right]. \tag{2.25}$$

These equations can be solved once initial conditions and boundary conditions have been specified. The basic process involves first solving (2.24) for $V^{n+\frac{1}{2}}(i)$ at all values of i. Once these values are known, they can be used in (2.25) to compute $I^{n+1}\left(i+\frac{1}{2}\right)$ for all values of i. This process is repeated within each complete Δt of the overall simulation. Due to this structure of feeding results back and forth between the equations, this kind of method is often referred to as using a *leap-frog time marching strategy.*

2.3 Stability Analysis: 1D Case

One common problem with time-marching methods is that it is often possible for a formulation to become *unstable* under certain conditions. In this context, *instability* refers to the numerical method producing exponentially growing solutions that quickly diverge to unbounded values regardless of the behavior of the simulation source. This kind of completely unphysical behavior must be suppressed for a numerical method to constitute a robust tool that can be used in practical applications.

We will now consider a standard approach for determining the stability of a time-stepping formula. The basic process involves expanding the quantity being solved (e.g., the voltage) in terms of a spatial Fourier series. The Fourier series representation can then be substituted into the time-stepping formula. Due to the orthogonality of the different Fourier modes, we can look at the energy in each of the modes independently as a function of time to determine whether it is possible for the method to become unstable.

As an example, we will consider the stability of (2.20). We begin by expanding the voltage in a spatial Fourier series as

$$V^{n}(i)=\sum_{m}A_{m}^{n}e^{jk_{m}i\Delta x},\quad k_{m}=\frac{m\pi}{L}, \tag{2.26}$$

where L is the length of the 1D computational domain and is given by $L=M\Delta x$. We now plug this representation into (2.20) with the source terms set to 0 (since we are interested in the intrinsic stability of the method) to get

$$\sum_{m}A_{m}^{n+1}e^{jk_{m}i\Delta x}=\sum_{m}\left[2(1-r)A_{m}^{n}-A_{m}^{n-1}+rA_{m}^{n}e^{-jk_{m}\Delta x}+rA_{m}^{n}e^{jk_{m}\Delta x}\right]e^{jk_{m}i\Delta x}, \tag{2.27}$$

where

$$r = \frac{(\Delta t)^2}{LC(\Delta x)^2}. \tag{2.28}$$

Due to the orthogonality of the Fourier modes, we can simplify this to

$$A_m^{n+1} = 2(1-r)A_m^n + 2r\cos(k_m \Delta x)A_m^n - A_m^{n-1}. \tag{2.29}$$

We can then use the trigonometric half-angle identities to further simplify this to be

$$A_m^{n+1} = 2[1 - 2r\sin^2(k_m \Delta x/2)]A_m^n - A_m^{n-1}. \tag{2.30}$$

We can now define a *amplification factor* for each mode as

$$g_m = \frac{A_m^{n+1}}{A_m^n} = \frac{A_m^n}{A_m^{n-1}}, \tag{2.31}$$

where the second equality follows from the fact that the same time-stepping formula is used in every time step of the simulation. By further defining

$$\alpha_m = 1 - 2r\sin^2(k_m \Delta x/2), \tag{2.32}$$

we can write (2.30) as

$$g_m^2 - 2\alpha_m g_m + 1 = 0. \tag{2.33}$$

The solution to this equation is

$$g_m = \alpha_m \pm \sqrt{\alpha_m^2 - 1}. \tag{2.34}$$

The time-stepping formula will only be stable if $|g_m| \leq 1$, which correspondingly requires $\alpha_m^2 \leq 1$. Considering this, we need to have

$$[1 - 2r\sin^2(k_m \Delta x/2)]^2 \leq 1. \tag{2.35}$$

Depending on the value of r, the maximum value of the left-hand side of (2.35) could occur for $(1-2r)^2$. From this, we see that to have the inequality in (2.35) satisfied requires

$$r = \frac{(\Delta t)^2}{LC(\Delta x)^2} \leq 1. \tag{2.36}$$

This can be rearranged as

$$\boxed{\Delta t \leq \Delta x\sqrt{LC} = \frac{\Delta x}{c},} \tag{2.37}$$

where it is recalled that for a transmission line $c = 1/\sqrt{LC}$ is the propagation speed on the line. We typically refer to (2.37) as the *stability condition* of the finite differencing method. This kind of "stability limit" is also often referred to as the *Courant-Fredrichs-Lewy (CFL)* *condition* for a time-stepping system. We also see from this analysis that this method is *conditionally stable*. That is, we cannot independently select the values for Δt and Δx – they must be selected in accordance with the stability condition for the method to provide useful results. From an intuitive perspective, we see that we must select Δt in such a way that we can resolve the propagation of signals from one grid point to the next.

It should also be noted that if we use different differencing formulas (e.g., backward or forward differences), our stability condition can change. It is possible to develop *unconditionally stable* methods, while it is also possible to develop *unconditionally unstable* methods. Typically, we develop methods using a combination of considerations related to accuracy and stability to determine which differencing formulas to use. For many traditional applications, central differences have been preferred because of their second-order accuracy and because they are usually conditionally stable. However, other applications can benefit from exploring other kinds of differencing formulas to optimize their performance for a particular application space.

If transmission lines with different parameters are used in the same simulation domain, the Δt that is used in the overall simulation should be the smallest one required by (2.37) for the different regions of the problem. This is one major drawback of the FDTD method. If a problem exists that requires a small Δx over a particular region of the problem it can force the time step of the overall simulation to be significantly smaller than would be required from a strictly sampling theory perspective. This can greatly increase the overall computation time of a method. It is an active area of research in developing more sophisticated kinds of finite differencing schemes that do not suffer as strongly from this kind of drawback.

In general, Δt and Δx are selected to obey (2.37) as close to "equality" as possible to lower the overall simulation time. However, a small safety factor is often included for practical situations, e.g., $\Delta t = 0.99\Delta x/c$. The particular problem being considered will dictate whether Δt or Δx should be selected first. On one hand, Δx must be selected so that it can properly resolve the spatial variation of the geometry being considered. If the geometry has features much smaller than the wavelength, this will often drive Δx to be much smaller than would be required purely from a temporal sampling perspective combined with (2.37). On the other hand, if we are interested in rather high frequency effects in our simulation we need to ensure that Δt is small enough to properly sample the temporal variations of the voltages. Typically, $\Delta t < T/20$, where T is the period of the highest frequency of interest in the simulation. We can then resolve any discrepancies between desired values for Δx or Δt by using (2.37).

2.4 Numerical Dispersion: 1D Case

Another important characteristic of finite differencing methods is the *numerical dispersion* that occurs in the solution process. This is a completely numerical error that is (typically) unavoidable within finite differencing methods. Briefly, numerical dispersion occurs because the simulated wave propagates with a velocity that deviates from the exact result due to the numerical discretization process. This can lead to an accumulation of "phase errors" in the solution process that can then impact the overall accuracy of simulated results.

One simple way to estimate the degree of numerical dispersion that will occur for a particular discretization scheme is to propagate a known signal with the time-stepping equations and compare this result to the exactly known analytical result. For a transmission line, the simplest wave to consider is a purely monochromatic wave propagating through a transmission line with constant electrical parameters. The analogue of this for a 3D system would be a monochromatic plane wave propagating through a homogeneous medium.

We will now consider a monochromatic voltage wave given by

$$V(x,t) = \mathrm{Re}\left\{ V_0 e^{j(\omega t - \beta x)} \right\}, \tag{2.38}$$

where $\beta = \omega\sqrt{LC}$ is the phase constant of the transmission line. On the finite difference grid, the simulated monochromatic voltage will be

$$V^n(i) = \mathrm{Re}\left\{ V_0 e^{j(\omega n\Delta t - \tilde{\beta} i \Delta x)} \right\}, \tag{2.39}$$

where $\tilde{\beta}$ is the numerical phase constant that we wish to compute to assess the numerical dispersion.

We can now plug (2.39) into our time-stepping formula of (2.20) and try to solve for $\tilde{\beta}$. This gives us

$$\mathrm{Re}\left\{ e^{j\omega\Delta t} = 2(1-r) + r\left[e^{j\tilde{\beta}\Delta x} + e^{-j\tilde{\beta}\Delta x}\right] - e^{-j\omega\Delta t} \right\} \tag{2.40}$$

after factoring out the common $\exp[j(\omega n\Delta t - \tilde{\beta} i \Delta x)]$ factor from the equation. Further, we have again simplified the notation by consolidating the various EM and discretization constants into r, which is given in (2.36). We can now simplify this equation to give us

$$\cos(\omega\Delta t) = (1-r) + r\cos(\tilde{\beta}\Delta x). \tag{2.41}$$

This can be readily solved for $\tilde{\beta}$ as

$$\boxed{\tilde{\beta} = \frac{1}{\Delta x}\arccos\left(1 - \frac{2}{r}\sin^2(\omega\Delta t/2)\right).} \tag{2.42}$$

This can be numerically compared to the exact phase constant to estimate the amount of numerical dispersion that will occur in a simulation.

To gain more insight into the behavior of the numerical dispersion, it can be valuable to derive an approximate expression that is easier to interpret than (2.42). This can be done by expanding the cosine terms in (2.41) using the first three terms of their Taylor series. Doing this, we get

$$1 - \frac{(\omega\Delta t)^2}{2} + \frac{(\omega\Delta t)^4}{24} \approx (1-r) + r\left[1 - \frac{(\tilde{\beta}\Delta x)^2}{2} + \frac{(\tilde{\beta}\Delta x)^4}{24}\right], \tag{2.43}$$

which can be simplified to

$$\beta^2 - \frac{1}{12}\beta^2(\omega\Delta t)^2 \approx \tilde{\beta}^2 - \frac{1}{12}\tilde{\beta}^2(\tilde{\beta}\Delta x)^2 \tag{2.44}$$

after expanding r back into its explicit values. We can now begin to rearrange this to get a kind of normalized error in the phase constant. To begin, we first rearrange the equation as

$$\frac{\tilde{\beta}^2-\beta^2}{\beta^2} \approx \frac{1}{12}\left[\frac{\tilde{\beta}^2}{\beta^2}(\tilde{\beta}\Delta x)^2-(\omega\Delta t)^2\quad\right]. \tag{2.45}$$

Next, we will make a few more approximations to simplify the result to something that can be easily evaluated. In particular, we will assume that on the left-hand side $\tilde{\beta}+\beta \approx 2\beta$ so that when we factor the numerator we have $\tilde{\beta}^2-\beta^2=(\tilde{\beta}-\beta)(\tilde{\beta}+\beta)\approx 2\beta(\tilde{\beta}-\beta)$. If we further assume on the right-hand side that $\tilde{\beta}\approx\beta$ then we can get our desired normalized error in the phase constant as

$$\boxed{\frac{\tilde{\beta}-\beta}{\beta} \approx \frac{1}{24}[(\beta\Delta x)^2-(\omega\Delta t)^2].} \tag{2.46}$$

If we were to choose $\Delta t = \Delta x/c$ (the maximum allowed by the stability condition), the error in the numerical phase constant would approximately vanish. This approximate vanishing of the numerical phase error can only occur in the simple 1D case. For the more general case of 2D and 3D analysis that we will consider later in this course, we will see that numerical dispersion will always be present for a general FDTD method. Now, even for the 1D case, it is still common to take $\Delta t < \Delta x/c$ so that there will be some non-vanishing error. This can lead to an accumulation of phase error according to

$$\frac{\text{Phase error}}{\lambda} \approx \frac{\tilde{\beta}-\beta}{\beta}\times\frac{360^\circ}{\lambda}, \tag{2.47}$$

where λ is the wavelength on the transmission line. If a simulation covers a large spatial extent or is ran for a long time, this numerical error can eventually corrupt the solution and make the results less reliable. However, from (2.46) and the stability condition, we can see that the numerical dispersion depends on $(\Delta x/\lambda)^2$. Hence, by reducing the spatial discretization size appropriately the numerical dispersion can be lowered.

Determining how fine of a grid size is necessary to produce accurate results can be a time consuming and difficult process. Usually, it is necessary to perform a kind of convergence study with a simulation method. In this convergence study, the problem is solved over and over again for increasingly fine discretizations until the output results of interest no longer change an appreciable amount in between two simulations. Many commercial CEM tools have this kind of convergence "study" built into them to try and prevent non-expert users from generating inaccurate simulation data. However, if you are developing your own code or are not careful with a commercial tool, it may become very important for you to perform a suitable convergence study yourself to meet the accuracy needs of your particular application.

2.5 Solution of 2D Electromagnetic Equations

Previously, we considered how the finite difference method could be used to discretize and solve 1D equations. We specifically focused on transmission lines, with both discretization methods we discussed corresponding to analyzing wave equations. We will now consider how the finite difference method can be used to handle 2D analysis. We will again focus on the wave equation initially, where we will see that the extension of the finite difference method to higher dimensions is relatively straightforward. The area that changes most significantly is the results of the stability and numerical dispersion analyses.

2.5.1 Equations and Discretization

We will begin by reviewing the equations applicable to 2D analysis. For this case to be relevant, we must consider the source and medium to be completely uniform along a particular axis, e.g., the z-axis. Due to this uniformity, none of the fields will have variation along the z-direction. We will further simplify the problem by assuming that as a source we only have an electric current that is flowing in the z-direction. This source can only produce an electric field that is also polarized along the z-axis. As a result, it suffices for us to develop an equation to solve only for the E_z component of the field.

To find our equation, we will start with Maxwell's curl equations with loss accounted for in terms of a conduction current. Note that the loss will be accounted for in this way as opposed to lumping this effect into the permittivity or permeability of the medium because loss in these material properties can only be handled in the time domain by treating the material as a dispersive medium, which greatly complicates the discretization of the mathematical system (the same is not true in the frequency domain, where loss can be accounted for more easily). We will return to this topic later in this course.

Now, considering all of these points, Maxwell's curl equations are

$$\nabla \times \mathbf{H} = \epsilon \partial_t \mathbf{E} + \sigma \mathbf{E} + \mathbf{J}_i, \tag{2.48}$$

$$\nabla \times \mathbf{E} = -\mu \partial_t \mathbf{H}, \tag{2.49}$$

where $\mathbf{J}_i$ is a given impressed current source (i.e., it is known *a priori* at all time values of interest and is not modified by the presence of the produced electromagnetic fields). These can be combined to form the vector wave equation for $\mathbf{E}$ by taking the curl of (4.66) and substituting in for $\nabla \times \mathbf{H}$ from (2.94). Performing this, we arrive at

$$\nabla \times \nabla \times \mathbf{E} + \mu\epsilon \partial_t^2 \mathbf{E} + \mu\sigma \partial_t \mathbf{E} = -\mu \partial_t \mathbf{J}_i. \tag{2.50}$$

Next, we can use

$$\nabla \times \nabla \times \mathbf{f} = \nabla(\nabla \cdot \mathbf{f}) - \nabla^2 \mathbf{f} \tag{2.51}$$

to rewrite (2.96) into the vector Helmholtz wave equation as

$$\nabla^2 \mathbf{E} - \mu\epsilon \partial_t^2 \mathbf{E} - \mu\sigma \partial_t \mathbf{E} = \mu \partial_t \mathbf{J}_i. \tag{2.52}$$

after noting that $\nabla \cdot \mathbf{E} = 0$. In a typical derivation of the wave equation we would rely on using Gauss' law to tell us that $\nabla \cdot \mathbf{E} = 0$. However, this does not apply directly here because we have an inhomogeneous medium so that Gauss' law actually gives us $\nabla \cdot \epsilon \mathbf{E} = 0$.

The reason $\nabla \cdot \mathbf{E} = 0$ here is simply because $\mathbf{E} = \hat{z}E_z$ and that there is no variation in the fields or properties along the z-direction of the problem.

We may now apply the specific knowledge about our problem to simplify this equation into a form more suitable for finite difference discretization. In particular, we will note that $\mathbf{E} = \hat{z}E_z$ and $\mathbf{J}_i = \hat{z}J_{i,z}$. By further recalling that there is no z-variation of the fields, we can expand the vector Laplacian operator into Cartesian components to give us

$$\partial_x^2 E_z + \partial_y^2 E_z - \mu\epsilon\partial_t^2 E_z - \mu\sigma\partial_t E_z = \mu\partial_t J_{i,z}. \tag{2.53}$$

We can now go about using our finite difference approximations to discretize this equation. As a starting point, we will break the problem up into a 2D grid with discretization sizes of Δx and Δy. Each point on the grid can be represented by two integers (i, j), which will correspond to the discrete position $(i\Delta x, j\Delta y)$. We will extend our shorthand notation with another spatial argument so that

$$E_z(i\Delta x, j\Delta y, n\Delta t) \to E_z^n(i, j). \tag{2.54}$$

We will also now keep better track of what values should be used for the different constitutive parameters in each $\Delta x \times \Delta y$ sized rectangular cell as

$$\epsilon(i\Delta x, j\Delta y) = \epsilon_{ij}, \tag{2.55}$$

with similar notation also used for μ and σ. Considering that these constitutive parameters are treated as taking a constant value within each rectangular cell, we achieve a discretization of our original problem as shown in Fig. 2.3. The constant values of constitutive parameters cause what is known as a *stair-casing approximation error.* This is one of the main drawbacks of the finite difference method, since the stair-casing errors can force one to use a very small discretization size for complex, curved surfaces. Further, the many sharp corners and edges in a stair-cased model can produce inaccurate physics, particularly close to a geometry. Due to these drawbacks, there have been many attempts over the years to develop improved discretization approaches to maintain the simplicity of finite difference methods while reducing the negative effects of stair-casing errors. A further example of stair-casing errors for a 3D geometry is shown in Fig. 2.4.

One approach to minimize the stair-casing error locally is to use a non-uniform FDTD grid where the size of the different cells changes throughout the simulation region. However, to maintain a regular rectangular shape to the cells that always lies along our different coordinate axes only a single cell dimension may be updated locally at a given step in the grid. This leads to "bands" in our mesh that require us to potentially still use excessively small cell sizes away from the areas we were attempting to refine (see Fig. 2.5). This can increase the computation time and produce other somewhat unpredictable effects (e.g., changing the numerical dispersion properties throughout the simulation region).

For our analysis here we will just use a regular grid that has no non-uniform refinement. With this grid ready, we can go about finding our time-stepping formula. To do this, we will apply central differencing formulas for all of the derivatives in (2.53). This process is no different from the 1D case; however, our algebraic equations become rather more tedious to

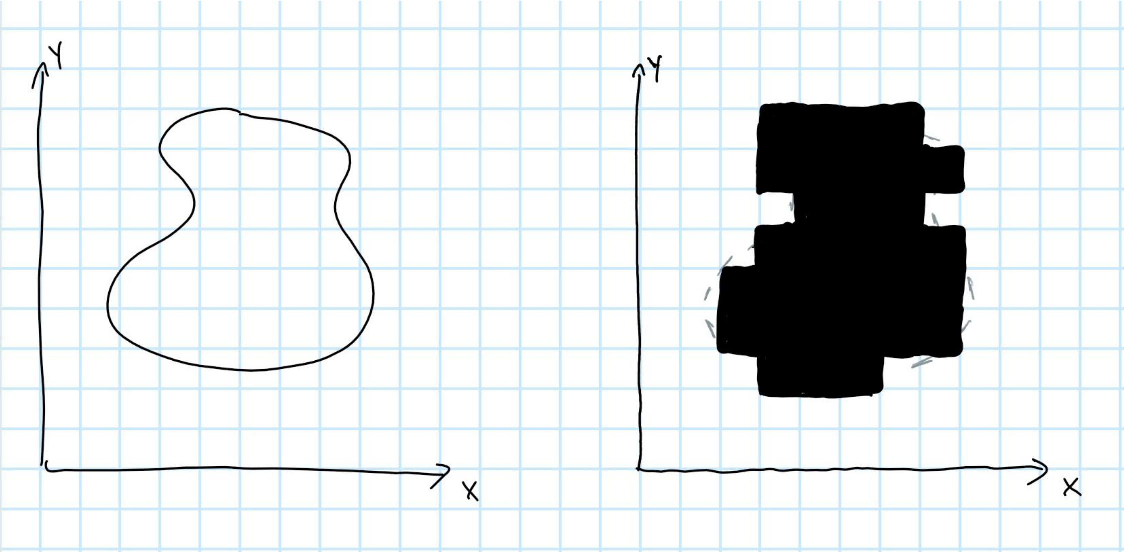
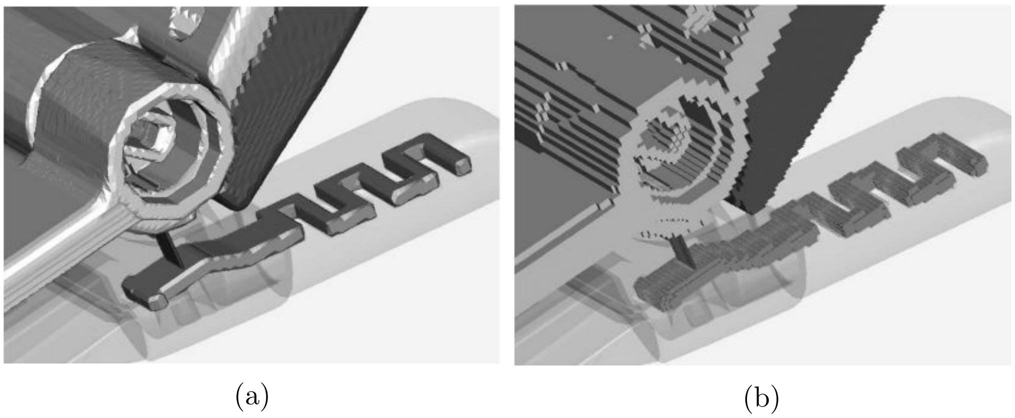


Figure 2.3: Discretization of a 2D scatterer. The image on the right shows the discrete representation with (severe) stair-casing approximation error.


(a)

(b)

Figure 2.4: Examples of (a) conformal and (b) stair-cased discretizations of an antenna near the joint of a flip phone (image from [3]). Note that this conformal mesh is generated to be used in an advanced FDTD method, which may be why it is still a relatively poor conformal mesh (alternatively, it could be limited by available computational resources at the time). Generally, better meshes are made when using more advanced CEM techniques such as the finite element method or the method of moments.

work with by hand. The result of the central differencing approximation is

$$\frac{E_z^n(i+1,j)-2E_z^n(i,j)+E_z^n(i-1,j)}{(\Delta x)^2}+\frac{E_z^n(i,j+1)-2E_z^n(i,j)+E_z^n(i,j-1)}{(\Delta y)^2}$$
$$-\mu_{ij}\epsilon_{ij}\frac{E_z^{n+1}(i,j)-2E_z^n(i,j)+E_z^{n-1}(i,j)}{(\Delta t)^2}-\mu_{ij}\sigma_{ij}\frac{E_z^{n+1}(i,j)-E_z^{n-1}(i,j)}{2(\Delta t)}$$
$$=\mu_{ij}\frac{J_{i,z}^{n+1}(i,j)-J_{i,z}^{n-1}(i,j)}{2(\Delta t)}. \quad (2.56)$$

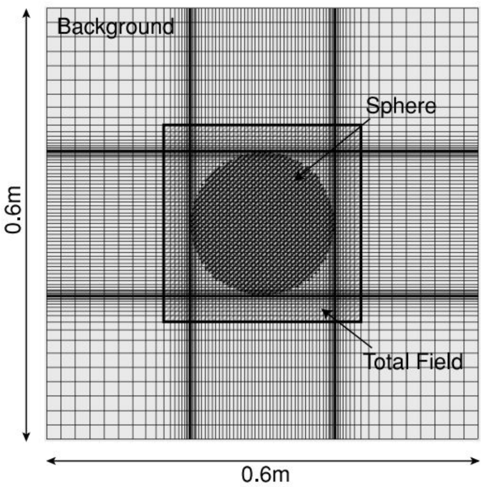


Figure 2.5: Example of how "local" mesh refinement in a non-uniform finite difference grid spreads throughout large regions of the simulation (image from [4]).

This can be rearranged into the following time-stepping formula:

$$E_z^{n+1}(i,j) = a_{ij}\Bigg\{2E_z^n(i,j)\left[\frac{\mu_{ij}\epsilon_{ij}}{(\Delta t)^2} - \frac{1}{(\Delta x)^2} - \frac{1}{(\Delta y)^2}\right] + b_{ij}E_z^{n-1}(i,j) + \frac{1}{(\Delta x)^2}\big[E_z^n(i+1,j) + E_z^n(i-1,j)\big] + \frac{1}{(\Delta y)^2}\big[E_z^n(i,j+1) + E_z^n(i,j-1)\big] - \frac{\mu_{ij}}{2\Delta t}\big[J_{i,z}^{n+1}(i,j) - J_{i,z}^{n-1}(i,j)\big]\Bigg\}, \quad (2.57)$$

where

$$a_{ij} = \left[\frac{\mu_{ij}\sigma_{ij}}{2\Delta t} + \frac{\mu_{ij}\epsilon_{ij}}{(\Delta t)^2}\right]^{-1}, \quad (2.58)$$

$$b_{ij} = \left[\frac{\mu_{ij}\sigma_{ij}}{2\Delta t} - \frac{\mu_{ij}\epsilon_{ij}}{(\Delta t)^2}\right]. \quad (2.59)$$

Although the expressions have grown considerably in size compared to the 1D case, the overall time-stepping process and many aspects of the code implementation do not need to change. As we will see when we consider other CEM methods later in the course, this simplicity in going to higher dimensions for numerical analysis is *uncommon*, and is one of the main attractive features of finite difference methods.

2.5.2 Stability Analysis: 2D Case

We will now consider the stability analysis of this set of equations. To simplify the process, we will assume that there is no loss so that $\sigma_{ij} = 0, \forall i, j$. We can now go about the same process as was used for the 1D case. However, we will now need to use a 2D spatial Fourier series so that

$$E_z^n(i,j) = \sum_{g,h} A_{g,h}^n e^{j(k_g i\Delta x + k_h j\Delta y)}, \; k_g = \frac{g\pi}{L_x}, \; k_h = \frac{h\pi}{L_y}, \tag{2.60}$$

where L_x and L_y are the lengths of the computational domain along the x- and y-directions, respectively. We can plug this representation into (2.57) to get

$$A_{g,h}^{n+1} = 2(1-r-s)A_{g,h}^n - A_{g,h}^{n-1} + r(e^{-jk_g\Delta x} + e^{jk_g\Delta x})A_{g,h}^n + s(e^{-jk_h\Delta y} + e^{jk_h\Delta y})A_{g,h}^n, \tag{2.61}$$

where

$$r = \frac{(\Delta t)^2}{\mu\epsilon(\Delta x)^2}, \tag{2.62}$$

$$s = \frac{(\Delta t)^2}{\mu\epsilon(\Delta y)^2}. \tag{2.63}$$

Now, we can rewrite the remaining exponential terms into cosine functions and then use the trigonometric half-angle identities to arrive at

$$A_{g,h}^{n+1} = 2\big[1 - 2r\sin^2(k_g\Delta x/2) - 2s\sin^2(k_h\Delta y/2)\big]A_{g,h}^n - A_{g,h}^{n-1}. \tag{2.64}$$

We can define the amplification factors as

$$G_{g,h} = \frac{A_{g,h}^{n+1}}{A_{g,h}^n} = \frac{A_{g,h}^n}{A_{g,h}^{n-1}} \tag{2.65}$$

and write the resulting polynomial equation into the same form as for the 1D case by defining

$$\alpha_{g,h} = 1 - 2r\sin^2(k_g\Delta x/2) - 2s\sin^2(k_h\Delta y/2). \tag{2.66}$$

The resulting polynomial equation is

$$G_{g,h}^2 - 2\alpha_{g,h}G_{g,h} + 1 = 0, \tag{2.67}$$

whose solution is

$$G_{g,h} = \alpha_{g,h} \pm \sqrt{\alpha_{g,h}^2 - 1}. \tag{2.68}$$

To ensure a stable time-marching method, we must have $\alpha_{g,h}^2 \leq 1$. Considering this, we need

$$[1 - 2r\sin^2(k_g\Delta x/2) - 2s\sin^2(k_h\Delta y/2)]^2 \leq 1. \tag{2.69}$$

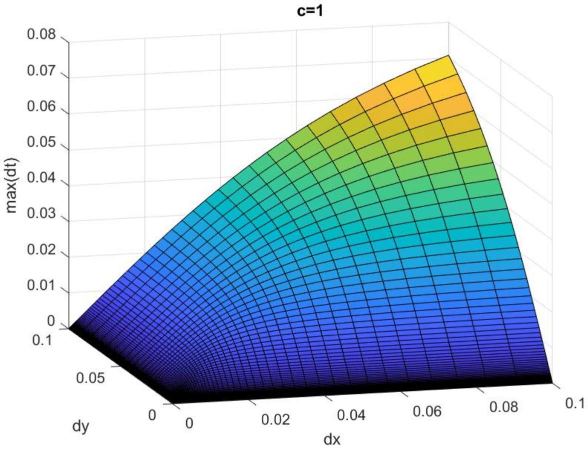


Figure 2.6: Change in stability condition as a function of Δx and Δy. Note that only modifying one of the discretization sizes can still strongly affect Δt.

The value of this inequality that we need to consider more carefully occurs when both $\sin^2$ terms equal 1, which gives the left-hand side of the inequality to be $(1-2r-2s)^2$. This then tells us that

$$r+s=\frac{(\Delta t)^2}{\mu\epsilon(\Delta x)^2}+\frac{(\Delta t)^2}{\mu\epsilon(\Delta y)^2}\leq 1. \tag{2.70}$$

We can rearrange this into our standard stability condition form as

$$\Delta t\leq\frac{1}{c\sqrt{\dfrac{1}{(\Delta x)^2}+\dfrac{1}{(\Delta y)^2}}}, \tag{2.71}$$

where we have introduced the speed of light as $c=1/\sqrt{\mu\epsilon}$. In choosing a time step for a simulation with inhomogeneous materials, the smallest acceptable time step should be selected based off of inserting the different constitutive parameters of the various parts of the simulation into (2.71).

One point to note about (2.71) is that if a single one of the discretization grid sizes has been shrunk to a smaller value to represent a particular geometry this will strongly lower the Δt that can be used to achieve stable simulation results. A plot of the variation of the maximum value for Δt as a function of Δx and Δy with $c=1$ (for simplicity at viewing the dependence of the function) is shown in Fig. 2.6.

2.5.3 Numerical Dispersion Analysis: 2D Case

In the previous section, we saw that extending our analysis to 2D resulted in a different stability condition for the resulting time-stepping formula. However, the analysis was able to cleanly separate its behavior independently along the different coordinate axes so that the results were still relatively simple to generate. The same kind of effect happens for the numerical dispersion analysis as well. Due to this, we will not go into depth on the derivation of the 2D results, but will instead focus mainly on the important consequences of the results.

Considering this, we begin by stating the exact numerical dispersion results for the 2D case. These are that

$$\left[\frac{1}{c\Delta t}\sin\left(\frac{\omega\Delta t}{2}\right)\right]^2 = \left[\frac{1}{\Delta x}\sin\left(\frac{\tilde{k}_x\Delta x}{2}\right)\right]^2 + \left[\frac{1}{\Delta y}\sin\left(\frac{\tilde{k}_y\Delta y}{2}\right)\right]^2, \tag{2.72}$$

where $\tilde{k}_x$ and $\tilde{k}_y$ are the numerical wavenumber along the x- and y-directions, respectively. We can follow a similar process to the 1D case and use a Taylor series to find a more intuitive approximate result to inspect the numerical dispersion. In particular, we get

$$\frac{\tilde{k}-k}{k} \approx \frac{1}{24}\left[(k\Delta x)^2\cos^4(\varphi) + (k\Delta y)^2\sin^4(\varphi) - (\omega\Delta t)^2\right], \tag{2.73}$$

where $\tilde{k}^2 = \tilde{k}_x^2 + \tilde{k}_y^2$ and φ is the angle that is made between the x-axis and the propagation direction of the plane wave used in the numerical dispersion analysis.

There are a few important points to be made about (2.73).

1. There is no choice of Δt that can be made to cancel out the numerical dispersion in all directions simultaneously. Numerical dispersion is simply guaranteed to occur in a finite difference method like this.

2. The severity of the numerical dispersion depends on the direction of propagation with respect to the discretization grid. This can lead to an uneven dispersion/distortion of waves as they propagate in different directions through a simulation region. As a result, it is important to use a reasonable discretization size and convergence study to ensure the accuracy of numerical results, particularly for complex geometries.

3. The numerical dispersion reduces quadratically as a function of the electrical discretization size (i.e., the grid size divided by the wavelength).

To help visualize the numerical dispersion error, the phase error per wavelength is plotted in Fig. 2.7 for a few different discretization sizes.

2.6 Finite Difference Solution of Poisson's Equation

We will now consider how to use the finite difference method to discretize another 2D electromagnetic equation. In particular, we will consider Poisson's equation. This equation is relevant to electrostatic systems, but can also be useful in analyzing certain aspects of waveguide problems. For instance, it can be shown that the fields of transverse electromagnetic (TEM) modes of a transmission line can be found by solving Poisson's equation. Here, we will focus on deriving Poisson's equation for the electrostatic case rather than for the waveguide problem.

To begin, we recall that Gauss' law of electricity is

$$\nabla \cdot \mathbf{D} = \rho. \tag{2.74}$$

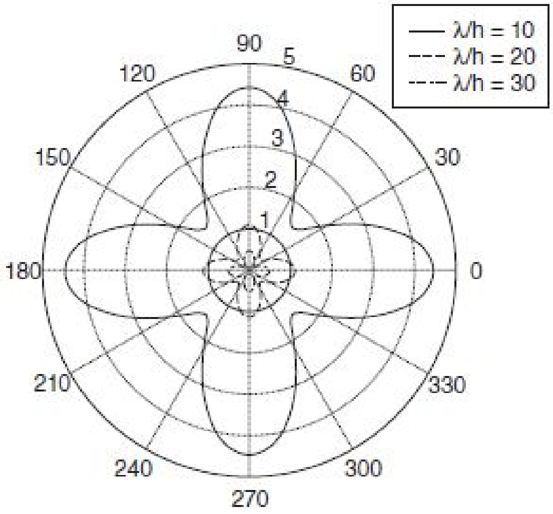


Figure 2.7: Numerical dispersion error as a function of wave propagation direction for $h = \Delta x = \Delta y$ (image from [5]). The error scale is in degrees per wavelength.

We can now use the constitutive relations to replace $\mathbf{D}$ with $\mathbf{E}$ to get

$$\nabla \cdot \epsilon \mathbf{E} = \rho. \tag{2.75}$$

Note that we have been careful to not factor the permittivity outside of the spatial derivative in this equation because we will be considering inhomogeneous regions later so that ϵ is not a constant function over all space. To arrive at Poisson's equation, we now recall that for electrostatic systems we can express $\mathbf{E}$ as the gradient of a scalar potential function (due to its curl-free nature). If we set $\mathbf{E} = -\nabla\phi$, we finally get Poisson's equation as

$$\nabla \cdot \epsilon \nabla \phi = -\rho. \tag{2.76}$$

To gain a little insight, let's begin by assuming that we are in a region of space where ϵ is homogeneous (i.e., constant). We can then move this to the right-hand side of the equation and then expand the scalar Laplacian in Cartesian coordinates to get

$$\partial_x^2\phi + \partial_y^2\phi = -\rho/\epsilon. \tag{2.77}$$

For simplicity, let's assume that we use the same discretization size for both x and y so that $\Delta x = \Delta y = h$. Then, we can use central differences to expand the two derivatives to get

$$\frac{\phi(i+1,j) - 2\phi(i,j) + \phi(i-1,j)}{h^2} + \frac{\phi(i,j+1) - 2\phi(i,j) + \phi(i,j-1)}{h^2} = -\rho/\epsilon. \tag{2.78}$$

We can rearrange this equation to solve for the value of ϕ at the current location (i,j). This gives us

$$\phi(i,j) = \frac{1}{4}\big[\phi(i+1,j) + \phi(i-1,j) + \phi(i,j+1) + \phi(i,j-1) + h^2\rho/\epsilon\big]. \tag{2.79}$$

We can recognize the right-hand side of this equation as essentially being an average of the surrounding potential values and the "data" at the current location (i.e., the value of the charge density modified by the permittivity). Hence, within a homogeneous region the Laplacian provides a kind of smoothing operation to the data.

If we inspect (2.79) closer, we can see that there is one major issue with it compared to our time-stepping equations. In particular, there are inter-dependencies in the data so that we cannot use a simple explicit "marching" solution process. As an alternative solution process, we can derive equations similar to (2.79) for each grid point of our simulation region and then assemble all of these equations into a matrix. We can then use a number of standard numerical linear algebra techniques to go about solving the overall Poisson's equation we set out to solve initially. We will discuss this in more detail at the end of this class. First, we will consider the more general situation of developing our equations when ϵ is no longer homogeneous.

2.6.1 Inhomogeneous Permittivity

As an example of a particular use case for a Poisson solver, we will now consider how to analyze the line capacitance of a microstrip transmission line. Due to the quasi-TEM nature of microstrip transmission lines, using a "static" Poisson solver can still provide valuable information about the field structure and characteristics of the dominant mode of the transmission line. The particular problem we will consider is shown in Fig. 2.8. To simplify the analysis, we consider a closed region that is formed by extending the ground plane of the microstrip line into a closed/"shielded" box. So long as the fictitious conductors are kept far away from the signal conductor of the transmission line, the overall potential distribution that is computed will be very close to the actual potential distribution that exists for a practical microstrip line that is not enclosed in a metal box.

To further simplify the analysis, we can exploit the *symmetry* of the problem. In particular, we can see that the structure is perfectly symmetric about the y-axis in Fig. 2.8. Considering this, we know that the solution of the potential throughout this entire problem will be mirrored/symmetric about the y-axis. Now, because the solution will be mirrored, we can quickly determine that the derivative in the x-direction of the potential along the symmetry plane will be identically 0. We can enforce this behavior as a boundary condition along the symmetry plane to avoid explicitly computing the potential within the entire geometry of Fig. 2.8. Instead, we can perform our analysis on the simplified geometry shown in Fig. 2.9 and get the same results by simply mirroring our solution across the symmetry plane at the end. This reduces the size of the computational region by half, which can lead to a much quicker solution (especially if the problem we are considering is large).

With our problem geometry outlined, we now need to go about solving for the potential in the region using the finite difference method. Since our problem is "excited" via the boundary conditions, we do not explicitly have a source term on the right-hand side of Poisson's equation (i.e., the impressed charge density is 0 everywhere). This special case is

$$\nabla \cdot \epsilon \nabla \phi = 0, \tag{2.80}$$

and is known as *Laplace's equation.*

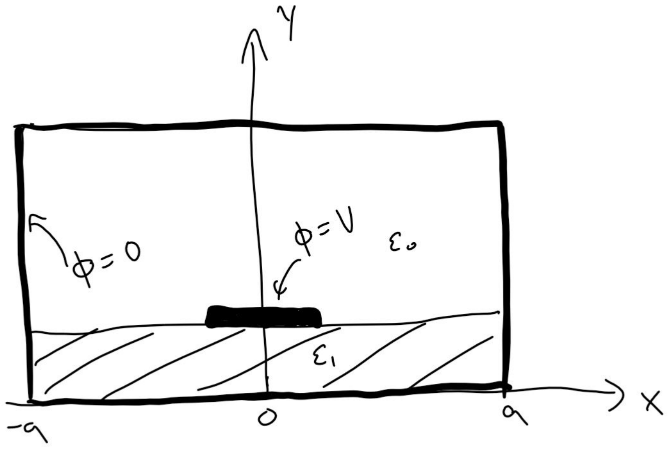


Figure 2.8: Depiction of the microstrip structure to be analyzed. As boundary conditions, a potential is placed on the signal conductor of the microstrip line and the ground plane and shield are set to $0\,V$.

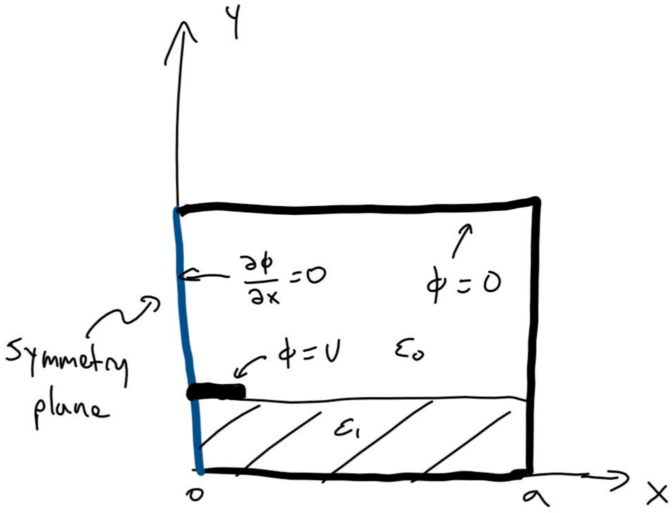


Figure 2.9: Depiction of the microstrip structure to be analyzed that uses a boundary condition to exploit the symmetry of the problem, reducing the size of the computational region by half.

Since we are working in Cartesian coordinates, we can readily expand the vector derivatives in this equation to see that we will need to find a finite difference representation for

$$\partial_x(\epsilon\partial_x\phi)+\partial_y(\epsilon\partial_y\phi)=0. \tag{2.81}$$

The main complication we will need to deal with here is the inhomogeneity in ϵ. In particular, we will need to make sure that we include it appropriately in the finite difference approximations.

To begin, we will focus on the x-derivatives. If we set $f(x, y) = \epsilon(x, y)\partial_x\phi(x, y)$, then we can use a central difference approximation to see that

$$\partial_x f(x, y) \approx \frac{f(x + h/2, y) - f(x - h/2, y)}{h}, \tag{2.82}$$

where we are continuing to assume that $h = \Delta x = \Delta y$ for simplicity. Now, we need to determine the explicit form for $f(x+h/2, y)$. We can expand this out and then use a central difference to get

$$\begin{aligned} f(x + h/2, y) &= \epsilon(x + h/2, y)\, \partial_x\phi(x + h/2, y) \\ &\approx \epsilon(x + h/2, y)\, \frac{\phi(x + h, y) - \phi(x, y)}{h}. \end{aligned} \tag{2.83}$$

Considering this, we can return to (2.82) to see that the complete result will be (after introducing our standard shorthand notation)

$$\begin{aligned} \partial_x f(x, y) &= \partial_x(\epsilon\partial_x\phi) \\ &= \frac{\epsilon(i + 1/2, j)}{h^2}[\phi(i + 1, j) - \phi(i, j)] - \frac{\epsilon(i - 1/2, j)}{h^2}[\phi(i, j) - \phi(i - 1, j)]. \end{aligned} \tag{2.84}$$

We can rewrite this into a more suggestive form as

$$\begin{aligned} \partial_x(\epsilon\partial_x\phi) = \frac{1}{h^2}\Big\{ &\epsilon(i + 1/2, j)\phi(i + 1, j) \\ &- \big[\epsilon(i + 1/2, j) + \epsilon(i - 1/2, j)\big]\phi(i, j) + \epsilon(i - 1/2, j)\phi(i - 1, j)\Big\}. \end{aligned} \tag{2.85}$$

A similar equation can be easily derived for the y-derivative term in (2.81), but will not be shown for brevity. From this equation, we see that our finite difference equations are naturally showing us that ϵ and ϕ lie on staggered grids with respect to each other, as shown in Fig. 2.10. However, when we go about discretizing a complex geometry it will not be uncommon for an ambiguity to occur; for example, how to decide what permittivity value to use when a point lies precisely at the interface between two regions with different permittivities. When this occurs, the general practice for finite difference methods is to compute the average of the nearby values of the quantity to approximate its value.

Now, there are a few special cases we need to consider carefully to ensure we develop correct finite difference formulas. The first is what happens when the point we are evaluating at is located precisely at the dielectric interface, as shown in Fig. 2.11. As mentioned previously, the general practice here will be to use an average of values when the permittivity is needed at data points that lie along the interface. Considering this, the finite difference approximation to (2.81) will become

$$\begin{aligned} \frac{1}{h^2}\bigg[&\frac{\epsilon_1 + \epsilon_0}{2}\Big(\phi(i + 1, j) - 2\phi(i, j) + \phi(i - 1, j)\Big) \\ &+ \epsilon_0\phi(i, j + 1) - (\epsilon_0 + \epsilon_1)\phi(i, j) + \epsilon_1\phi(i, j - 1)\bigg] = 0 \end{aligned} \tag{2.86}$$

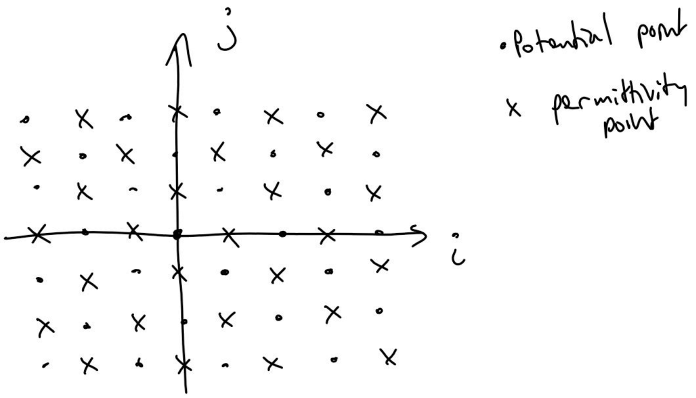


Figure 2.10: Staggered grid points where the finite difference method needs to sample the permittivity and potential at.

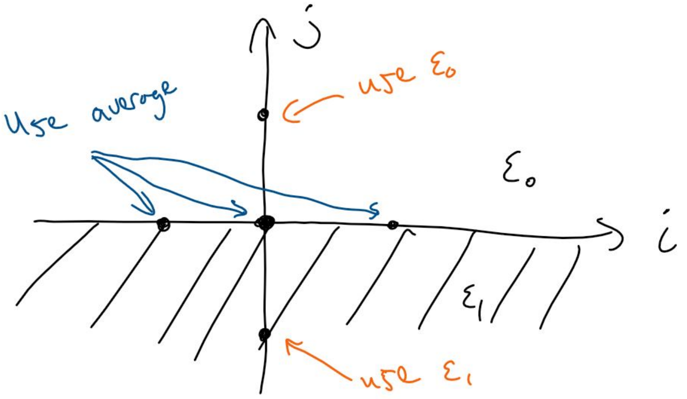


Figure 2.11: Evaluation of finite difference formulas at the inhomogeneous dielectric interface.

for the situation illustrated in Fig. 2.11.

The next special cases to consider are what happens toward the edges of the geometry when we encounter one of our boundary conditions. We will consider the Dirichlet boundary condition (which specifies the potential) first. An example of this situation is shown in Fig. 2.12, where we are specifically considering the case of a point at the maximum value of our x-grid that needs to be solved for. This point is denoted as (I, j), where $i \in [0, I]$. Here, (2.81) becomes

$$\frac{\epsilon_0}{h^2}\Big[\phi(I+1,j)+\phi(I-1,j)+\phi(I,j+1)+\phi(I,j-1)-4\phi(I,j)\Big]=0. \tag{2.87}$$

To format this so it can be easily incorporated into a matrix equation, we need to move the known quantity to the right-hand side of the equation. For this particular case, we know from our Dirichlet boundary condition that $\phi(I+1,j)=\varphi_0$, where φ_0 is the potential

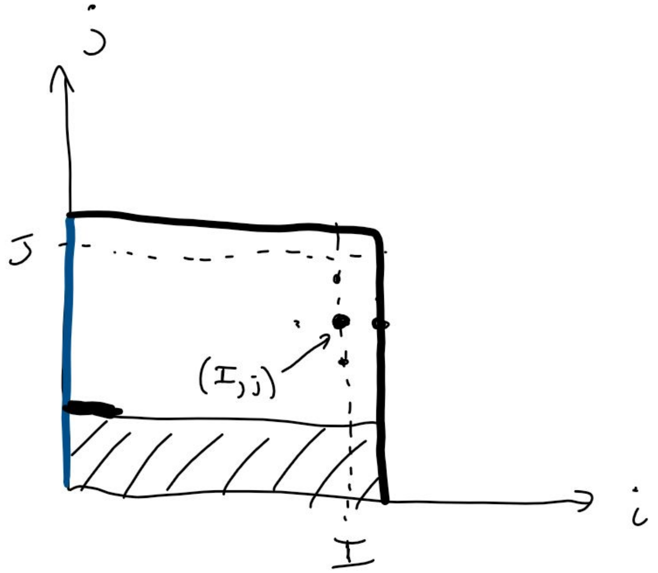


Figure 2.12: Evaluation of finite difference formulas near a Dirichlet boundary condition.

specified by the boundary condition. Hence, this equation becomes

$$\left[\phi(I-1,j)+\phi(I,j+1)+\phi(I,j-1)-4\phi(I,j)\right]=-\varphi_0. \tag{2.88}$$

For Fig. 2.12, we would have $\varphi_0 = 0$. We show the full formula here in (2.88) for clarity.

The final special case to consider is what happens at the symmetry plane of the problem where we have assigned a homogeneous Neumann boundary condition, as shown in Fig. 2.13. For this case, our boundary condition is that

$$\partial_x \phi|_{x=0} = 0. \tag{2.89}$$

We can expand this using a central difference approximation to determine that

$$\phi(-1,j) = \phi(1,j). \tag{2.90}$$

Hence, we can simplify (2.81) from

$$\phi(1,j)+\phi(-1,j)+\phi(0,j+1)+\phi(0,j-1)-4\phi(0,j)=0 \tag{2.91}$$

to

$$2\phi(1,j)+\phi(0,j+1)+\phi(0,j-1)-4\phi(0,j)=0. \tag{2.92}$$

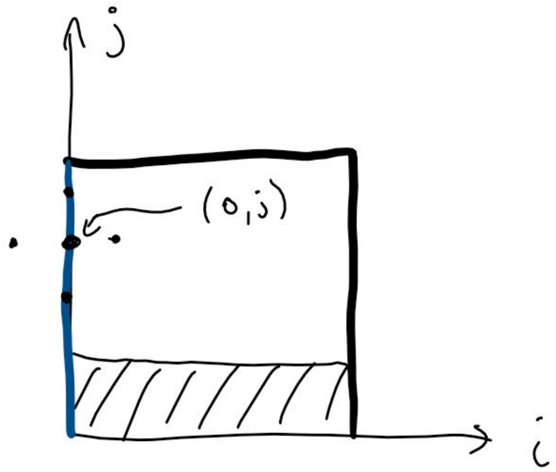


Figure 2.13: Evaluation of finite difference formulas near a Neumann boundary condition.

2.6.2 Matrix Equation Solution

As mentioned previously, the finite difference discretization of Laplace's equation did not lead to a kind of time- or space-marching formula that we can solve explicitly. Instead, the inter-dependencies between equations has left us with a linear system of equations. The typical strategy to solve this kind of problem is to assemble all of our equations into a matrix equation of the form

$$\overline{\mathbf{A}}\mathbf{x} = \mathbf{b}, \tag{2.93}$$

where $\overline{\mathbf{A}}$ will be built from the finite difference equations like (2.86), (2.88), and (2.92). The right-hand side vector **b** will be predominantly empty for the microstrip analysis considered in this section. However, it will take on non-zero values according to the Dirichlet boundary condition assigned to the signal conductor of the microstrip transmission line.

With the matrix equation determined, it can be solved using a number of techniques from numerical linear algebra. The most straightforward approaches are usually referred to as *direct solvers*. These either explicitly (or in essence) compute the inverse of the matrix. Examples of direct solvers are Gaussian elimination routines, performing an LU decomposition, or applying a specialized sparse direct solver. The straightforward direct solver techniques such as Gaussian elimination or LU decompositions are typically only possible for solving relatively small problems. The reason for this is the order of operations required to complete the numerical routine for these simple approaches is typically $O(N^3)$, where N is the number of elements in the matrix (equal to the number of discretization grid points for our current example). This scaling can quickly lead to inordinate computation times that are completely unacceptable. Hence, there is significant research devoted to improving the speed of direct solvers for specialized problem sets (such as the sparse matrices that are generated from finite difference and finite element methods).

An alternative to a direct solver is to use an *iterative solver*. These methods can be viewed as being similar to a kind of optimization routine. They begin by guessing a trial solution $\mathbf{x}_0$ and then compute $\overline{\mathbf{A}}\mathbf{x}_0$. The residual error is computed as $\mathbf{r} = \overline{\mathbf{A}}\mathbf{x} - \mathbf{b}$. This residual error is then used to update the trial solution to some new guess $\mathbf{x}_1$. The exact way this residual error is utilized depends on the particular iterative solver employed, of which there are many options available. This process of using the residual error to improve our trial solution continues until we ideally reach some level of *convergence* (i.e., the residual error drops below a specified value). One of the main benefits of this process is that the main computational bottleneck of these methods is computing the matrix-vector product $\overline{\mathbf{A}}\mathbf{x}_n$. In a worst-case scenario (i.e., a completely dense matrix), this operation can be completed in $O(N^2)$ steps, which can lead to a huge time-saving compared to a direct solver. When working with sufficiently sparse matrices (like those made from a finite difference method), the cost of a matrix-vector product can typically be completed in $O(N)$ operations, making iterative solvers a very useful option for solving large-scale simulation problems. However, many EM physics problems lead to matrix equations that are particularly difficult to solve iteratively. For these problems, the convergence may be very slow (which requires many iterations) or in extreme cases may not be able to converge at all. As a result, there is significant research devoted to developing improved discretization approaches that lead to more well-behaved matrix equations so that iterative solvers can be used successfully. One particularly important challenge associated with this kind of approach is ensuring that the sparsity of the matrices used is maintained so that iterative solvers can still be sufficiently efficient.

2.6.3 Post-Processing

Assuming we have successfully solved the matrix equation, we now have access to the potential at each point in our computational domain. We can now use this knowledge in various *post-processing* steps to learn more about the operation of our device. For instance, we can plot the potential to gain a visual understanding of how it varies throughout the computational domain. Another option would be numerically computing the gradient of the potential using finite difference formulas to compute the electric field. It is often a very valuable step to generate plots such as these to help us see if the solution matches our expectations for the geometry being considered. If there is a large difference between our expectations and the numerical solution, we may need to update our thought process or it could be a sign that our numerical solution was not completed correctly. This can happen frequently when we are first developing our own CEM code. However, it can also happen when we use mature, commercial CEM tools if we potentially have an error in how we set up our simulation parameters. As a result, these kinds of "sanity checks" are vital in checking the results produced by a CEM tool.

2.7 Finite Difference Discretization of the 3D Wave Equation

We will now briefly look at the finite difference discretization of the 3D wave equation to see the complications that arise for this situation. Due to these issues, this approach is relatively unpopular. Instead, a method known as Yee's FDTD scheme is typically used. We will discuss this scheme in the coming sections.

We will begin with Maxwell's curl equations, which are

$$\nabla \times \mathbf{H} = \epsilon \partial_t \mathbf{E} + \sigma \mathbf{E} + \mathbf{J}_i, \tag{2.94}$$

$$\nabla \times \mathbf{E} = -\mu \partial_t \mathbf{H}, \tag{2.95}$$

where $\mathbf{J}_i$ is an impressed current source. These can be combined to form the vector wave equation for $\mathbf{E}$ by taking the curl of (4.66) and substituting in for $\nabla \times \mathbf{H}$ from (2.94). Performing this, we arrive at

$$\nabla \times \mu^{-1} \nabla \times \mathbf{E} + \epsilon \partial_t^2 \mathbf{E} + \sigma \partial_t \mathbf{E} = -\partial_t \mathbf{J}_i. \tag{2.96}$$

We can then go about reducing (2.96) into 3 scalar equations and using our regular finite difference approximations. In particular, we will represent E_z on discrete grid points as

$$E_z(x, y, z, t) \to E_z^n(i, j, k) = E_z(i\Delta x, j\Delta y, k\Delta z, n\Delta t), \tag{2.97}$$

with similar representations for E_x and E_y. We can then apply central difference approximations to all of the derivatives and derive time-stepping equations for E_x, E_y, and E_z in a manner analogous to the 2D case.

Although this all appears fine, issues begin to arise when we want to consider how to implement our finite difference approximation near interfaces between regions with different constitutive parameters. Imagine we need to consider a grid point that lies precisely at the interface between two regions with different permittivities, as shown in Fig. 2.14. If the normal vector to the interface is oriented along the z-direction, we do not run into significant difficulty in using E_x or E_y. The reason for this is that the tangential component of the electric field is known to be continuous at the interface between two dielectric media, and so E_x and E_y do not have an ambiguity in their definitions at this point. However, when we consider the E_z equation we come to the issue that E_z is discontinuous across this dielectric interface (since only the normal component of the electric flux is continuous). Hence, it becomes ambiguous how E_z should be represented at this point.

Even more severe issues occur when the field grid point lies on the edge or corner of a dielectric or conductor. The issue is that field singularities can occur at these points, and the boundary conditions can again take on ambiguous meanings. The underlying cause of these problems lies in the fact that we are attempting to represent the electric field at *discrete points*. This is a fundamentally flawed way to think of the electric field in a mathematical sense, although we will not have time to go into the full mathematical formalism that shows this in this course (this mathematical formalism is related to the theory of *differential forms*, which have grown in popularity in the physics community to handle complicated gauge theories, such as Yang-Mills theory).

Instead of treating the electric (or magnetic) field as existing at discrete points, it is necessary to always think of it as something that is built to be integrated along a 1D curve (think about the units for the electric and magnetic fields). This notion is elegantly encapsulated in Yee's FDTD scheme, and is central to its success at resolving all of the issues with the method we discussed in this section (some details on how to use Yee's FDTD scheme to successfully discretize an equation similar to (2.96) can be found in [6]).

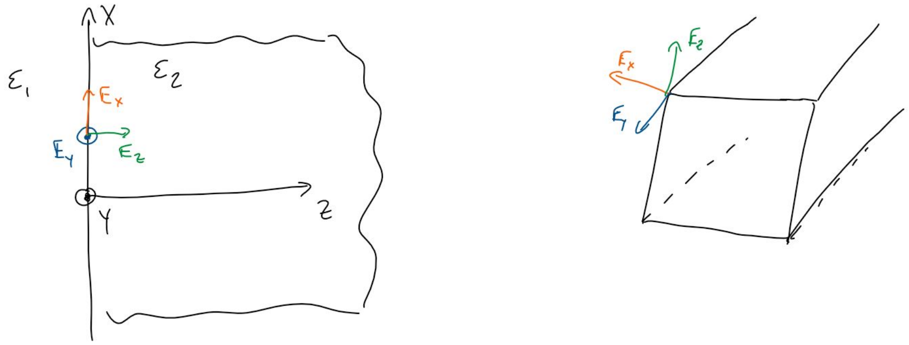


Figure 2.14: Situations where ambiguities/issues can arise in the finite difference discretization of the 3D wave equation. (Left) Grid points near an interface between two different materials and (right) grid points along edges and corners of an object.

2.8 Yee's FDTD Scheme – 2D Case

Yee's method involves solving Maxwell's curl equations as a set of coupled first-order partial differential equations. This leads to a method that is similar to the leap-frog time-marching method we considered for analyzing transmission lines using the telegrapher's equations. Similar to this case, Yee's method will end up using a staggered spatial and temporal grid. However, one of the main contributions of Yee's scheme was to transition from considering field data at discrete grid points to treating the data as a constant vector along an *edge* of the grid. As we will see, this resolves the issues with interfaces between different regions and has resulted in Yee's method being one of the most successful and widely used finite difference methods for performing numerical electromagnetic analysis.

To begin to understand Yee's method, we will start with the 2D case. As with our previous 2D analysis, we will assume that the problem geometry is completely uniform along the z-direction. We will only consider the case for a z-polarized electric field here, but will note in passing that this method can be easily reformulated to consider alternative polarization cases as well.

Now, for this 2D analysis we can reduce Maxwell's curl equations to the following set of three scalar equations:

$$\partial_y E_z = -\mu \partial_t H_x, \tag{2.98}$$

$$\partial_x E_z = \mu \partial_t H_y, \tag{2.99}$$

$$\partial_x H_y - \partial_y H_x = \epsilon \partial_t E_z + \sigma E_z + J_{i,z}. \tag{2.100}$$

We can now go about solving for E_z, H_x, and H_y by breaking up our computational domain into a set of rectangular cells. The center of each cell will correspond to locations where E_z is sampled at and will be identified with the integer pair (i, j). The magnetic field components

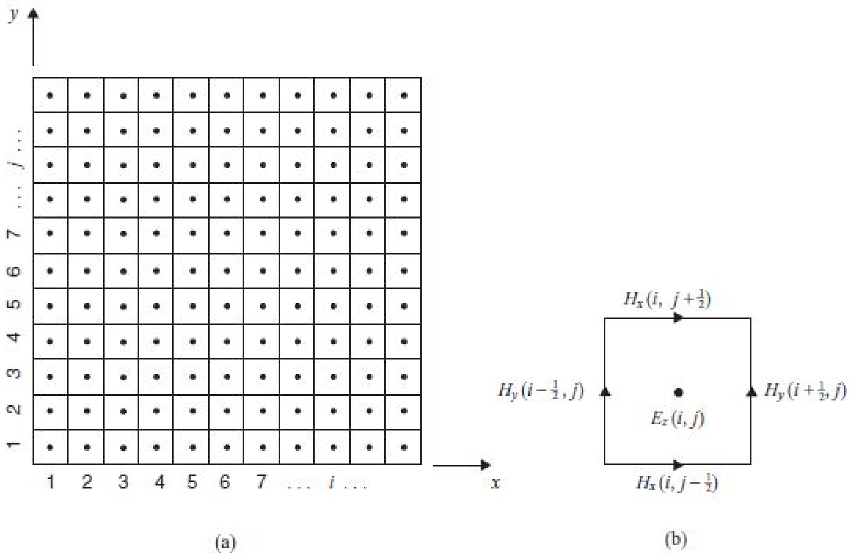


Figure 2.15: (a) Finite difference discretization for Yee's scheme in 2D and (b) locations of field components for a particular FDTD cell (images from [5]).

will be sampled along the *edges* of the rectangular cell that $E_z(i,j)$ is associated with. In particular, we will have H_x sampled along the x-directed edges located at $(i, j+1/2)$ and $(i, j-1/2)$ and H_y will be sampled along the y-directed edges located at $(i+1/2, j)$ and $(i-1/2, j)$. This setup is illustrated in Fig. 2.15. Note that due to this assignment of the magnetic field along these edges, the magnetic field will always be tangential to any interface that this edge lies along. As a result, the magnetic field will be well-defined due to the continuity of this field component at the interface.

The final piece of the discretization involves the temporal sampling points for the different field components. Following the example we saw for the leap-frog method for transmission lines, we will need to sample E_z at $t = n\Delta t$ and the magnetic field components will be sampled at half-integer values like $t = (n+1/2)\Delta t$.

Using this setup, we can use central differences to approximate (2.98) as

$$\frac{E_z^n(i,j+1) - E_z^n(i,j)}{\Delta y} = -\mu(i,j+1/2)\frac{H_x^{n+1/2}(i,j+1/2) - H_x^{n-1/2}(i,j+1/2)}{\Delta t}. \quad (2.101)$$

This can be rearranged into a time-stepping formula

$$H_x^{n+1/2}(i,j+1/2) = H_x^{n-1/2}(i,j+1/2) - \frac{\Delta t}{\mu(i,j+1/2)\Delta y}\left[E_z^n(i,j+1) - E_z^n(i,j)\right]. \quad (2.102)$$

A similar time-stepping formula can be found for H_y, and is

$$H_y^{n+1/2}(i+1/2,j) = H_y^{n-1/2}(i+1/2,j) + \frac{\Delta t}{\mu(i+1/2,j)\Delta x}\left[E_z^n(i+1,j) - E_z^n(i,j)\right]. \quad (2.103)$$

Finally, the time-stepping formula can be found for E_z, and is

$$\begin{aligned} E_z^{n+1}(i,j) = a(i,j)\Bigg\{ b(i,j)E_z^n(i,j) &+ \frac{1}{\Delta x}\Big[H_y^{n+1/2}(i+1/2,j) - H_y^{n+1/2}(i-1/2,j)\Big] \\ &- \frac{1}{\Delta y}\Big[H_x^{n+1/2}(i,j+1/2) - H_x^{n+1/2}(i,j-1/2)\Big] - J_{i,z}^{n+1/2}(i,j)\Bigg\}, \end{aligned} \tag{2.104}$$

where

$$a(i,j) = \left[\frac{\epsilon(i,j)}{\Delta t} + \frac{\sigma(i,j)}{2}\right]^{-1}, \tag{2.105}$$

$$b(i,j) = \left[\frac{\epsilon(i,j)}{\Delta t} - \frac{\sigma(i,j)}{2}\right]. \tag{2.106}$$

With suitable initial conditions and boundary conditions, we can follow a leap-frog time stepping scheme to progressively solve for E_z^{n+1}, which can then be used to solve for $H_x^{n+3/2}$ and $H_y^{n+3/2}$. A stability and numerical dispersion analysis can be completed for this scheme, which shows that the same results occur here as for what was found with the 2D analysis of the wave equation.

2.9 Yee's FDTD Scheme – 3D Case

Extending Yee's scheme to 3D follows relatively quickly from the 2D case. The main concept to grasp is how to properly visualize the *Yee grid* that the electric and magnetic field components are defined on. Once this is understood, deriving the time-stepping equations and implementing the code can be performed quite easily.

As with the 2D case, our starting point will be Maxwell's curl equations. These are

$$\nabla \times \mathbf{E} = -\mu\partial_t\mathbf{H}, \tag{2.107}$$

$$\nabla \times \mathbf{H} = \epsilon\partial_t\mathbf{E} + \sigma\mathbf{E} + \mathbf{J}_i. \tag{2.108}$$

These two vector equations can be easily expanded into six scalar equations that can then be converted to time-stepping equations. In particular, we will need to discretize

$$\partial_y E_z - \partial_z E_y = -\mu\partial_t H_x, \tag{2.109}$$

$$\partial_z E_x - \partial_x E_z = -\mu\partial_t H_y, \tag{2.110}$$

$$\partial_x E_y - \partial_y E_x = -\mu\partial_t H_z, \tag{2.111}$$

$$\partial_y H_z - \partial_z H_y = \epsilon\partial_t E_x + \sigma E_x + J_{i,x}, \tag{2.112}$$

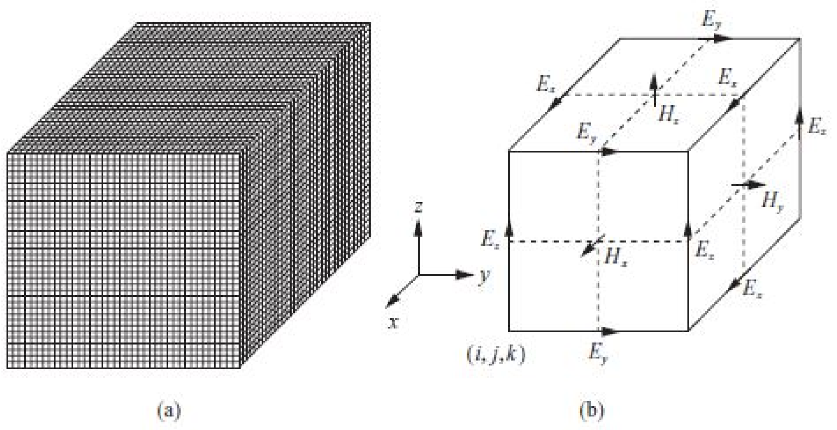


Figure 2.16: (a) Finite difference discretization for Yee's scheme in 3D and (b) locations of field components for a particular FDTD cell (images from [5]).

$$\partial_z H_x - \partial_x H_z = \epsilon \partial_t E_y + \sigma E_y + J_{i,y}, \tag{2.113}$$

$$\partial_x H_y - \partial_y H_x = \epsilon \partial_t E_z + \sigma E_z + J_{i,z}. \tag{2.114}$$

This discretization can be performed by dividing a volume of interest into small rectangular cells of size $\Delta x \times \Delta y \times \Delta z$. The electric field components are then assigned at the center of each *edge* of the rectangular cell that is parallel to the field vector. Although this is the "sampling point", the field is treated as a constant vector along the entire edge in a manner similar to how the magnetic field was treated in the 2D Yee method discussed in the previous section. The magnetic field components are then placed at the center of each face of the rectangular cell, as shown in Fig. 2.16. In reality, it is better to think of the magnetic field as being defined in the same manner as the electric field but along a *dual grid* that is offset from the electric field grid by a half grid size in each dimension (see Fig. 2.17). This discretization scheme does a good job of preserving the *duality* between the electric and magnetic fields by treating them both in identical manners. As with the 2D case, the time steps that the magnetic and electric field components are solved at will also be offset by each other by a half grid point.

Using central differences on the Yee grid, the magnetic field time-stepping formulas can be found to be

$$\begin{aligned} H_x^{n+1/2}(i,j+1/2,k+1/2) &= H_x^{n-1/2}(i,j+1/2,k+1/2) \\ &\quad - \frac{\Delta t}{\mu(i,j+1/2,k+1/2)\Delta y}\Big[E_z^n(i,j+1,k+1/2) - E_z^n(i,j,k+1/2)\Big] \\ &\quad + \frac{\Delta t}{\mu(i,j+1/2,k+1/2)\Delta z}\Big[E_y^n(i,j+1/2,k+1) - E_y^n(i,j+1/2,k)\Big], \end{aligned} \tag{2.115}$$

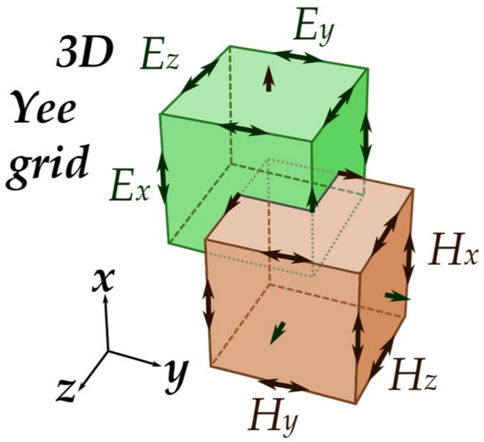


Figure 2.17: Illustration of the staggered grid interpretation of the Yee grid (image from [7]).

$$\begin{aligned} H_y^{n+1/2}(i+1/2,j,k+1/2) &= H_y^{n-1/2}(i+1/2,j,k+1/2) \\ &\quad - \frac{\Delta t}{\mu(i+1/2,j,k+1/2)\Delta z}\Big[E_x^n(i+1/2,j,k+1) - E_x^n(i+1/2,j,k)\Big] \\ &\quad + \frac{\Delta t}{\mu(i+1/2,j,k+1/2)\Delta x}\Big[E_z^n(i+1,j,k+1/2) - E_z^n(i,j,k+1/2)\Big], \end{aligned} \tag{2.116}$$

$$\begin{aligned} H_z^{n+1/2}(i+1/2,j+1/2,k) &= H_z^{n-1/2}(i+1/2,j+1/2,k) \\ &\quad - \frac{\Delta t}{\mu(i+1/2,j+1/2,k)\Delta x}\Big[E_y^n(i+1,j+1/2,k) - E_y^n(i,j+1/2,k)\Big] \\ &\quad + \frac{\Delta t}{\mu(i+1/2,j+1/2,k)\Delta y}\Big[E_x^n(i+1/2,j+1,k) - E_x^n(i+1/2,j,k)\Big]. \end{aligned} \tag{2.117}$$

A similar process can be performed for the electric field. This results in time-stepping formulas of

$$\begin{aligned} E_x^{n+1}(i+1/2,j,k) &= a(i+1/2,j,k)\Big\{b(i+1/2,j,k)E_x^n(i+1/2,j,k) \\ &\quad + \frac{1}{\Delta y}\Big[H_z^{n+1/2}(i+1/2,j+1/2,k) - H_z^{n+1/2}(i+1/2,j-1/2,k)\Big] \\ &\quad - \frac{1}{\Delta z}\Big[H_y^{n+1/2}(i+1/2,j,k+1/2) - H_y^{n+1/2}(i+1/2,j,k-1/2)\Big] \\ &\quad - J_{i,x}^{n+1/2}(i+1/2,j,k)\Big\}, \end{aligned} \tag{2.118}$$

$$\begin{aligned} E_y^{n+1}(i,j+1/2,k) = a(i,j+1/2,k)\Bigg\{ & b(i,j+1/2,k)E_y^{n}(i,j+1/2,k) \\ & + \frac{1}{\Delta z}\Big[H_x^{n+1/2}(i,j+1/2,k+1/2) - H_x^{n+1/2}(i,j+1/2,k-1/2)\Big] \\ & - \frac{1}{\Delta x}\Big[H_z^{n+1/2}(i+1/2,j+1/2,k) - H_z^{n+1/2}(i-1/2,j+1/2,k)\Big] \\ & - J_{i,y}^{n+1/2}(i,j+1/2,k)\Bigg\}, \end{aligned} \tag{2.119}$$

$$\begin{aligned} E_z^{n+1}(i,j,k+1/2) = a(i,j,k+1/2)\Bigg\{ & b(i,j,k+1/2)E_z^{n}(i,j,k+1/2) \\ & + \frac{1}{\Delta x}\Big[H_y^{n+1/2}(i+1/2,j,k+1/2) - H_y^{n+1/2}(i-1/2,j,k+1/2)\Big] \\ & - \frac{1}{\Delta y}\Big[H_x^{n+1/2}(i,j+1/2,k+1/2) - H_x^{n+1/2}(i,j-1/2,k+1/2)\Big] \\ & - J_{i,z}^{n+1/2}(i,j,k+1/2)\Bigg\}, \end{aligned} \tag{2.120}$$

where

$$a(i,j,k) = \left[\frac{\epsilon(i,j,k)}{\Delta t} + \frac{\sigma(i,j,k)}{2}\right]^{-1}, \tag{2.121}$$

$$b(i,j,k) = \left[\frac{\epsilon(i,j,k)}{\Delta t} - \frac{\sigma(i,j,k)}{2}\right]. \tag{2.122}$$

These time-stepping equations can be used in a leap-frog process to continue to advance the simulation forward in time.

As is common when working with a "staggered grid" for performing a finite difference discretization, it is possible for interfaces between regions to occur at "awkward" locations in one of the grids that makes it not immediately obvious which parameters should be used in the time-stepping equations. The typical wisdom is to use a relevant average of values for the quantity of interest (e.g., the permittivity or permeability). A more rigorous way to go about formulating the time-stepping equations is to utilize the *integral form of Maxwell's equations* [2].

It can be shown quite easily that the Yee method time-stepping equations can be derived from the integral form of Maxwell's equations applied to the Yee grid shown in Fig. 2.16. These simple techniques can be applied to the "awkward" discretization situations to derive how the time-stepping equations should be augmented due to the presence of inhomogeneities. It is often found that the "correct" way to augment the time-stepping equations is to simply use the average value of the relevant material parameters involved in the inhomogeneity [2].

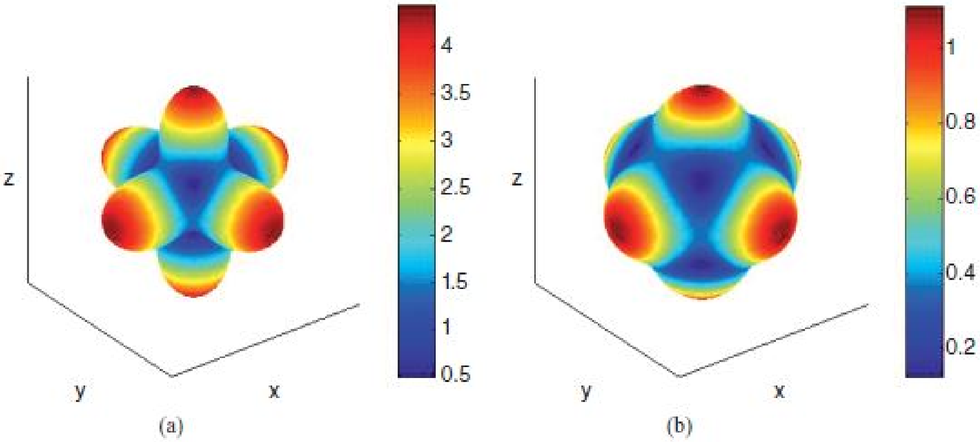


Figure 2.18: Illustration of the numerical phase error (in degrees) for the 3D Yee scheme for a mesh density of (a) $h/\lambda = 1/10$ and (b) $h/\lambda = 1/20$, where $h = \Delta x = \Delta y = \Delta z$ (images from [5]).

Although this averaging is typically sufficient to maintain second-order accuracy, to ensure the necessary continuities of various fields and fluxes does require us to align the material property definitions with various grids. In particular, the permittivity should always be aligned with the primary grid surfaces and the permeability should always be aligned with the dual grid surfaces. As a result, if a material has both electric and magnetic properties it will be discretized with a somewhat inaccurate boundary due to the staggering of the material properties starting by a half grid cell [2].

As with the other FDTD schemes we have discussed, the Yee method is subject to a stability condition. The analysis is somewhat tedious, but can follow the similar process to what we have done previously. The end result is a straightforward extension of the 2D results we established already. In particular, we get that

$$\Delta t \leq \frac{1}{c\sqrt{\dfrac{1}{(\Delta x)^2} + \dfrac{1}{(\Delta y)^2} + \dfrac{1}{(\Delta z)^2}}}. \tag{2.123}$$

Similarly, a numerical dispersion analysis can be performed to see that

$$\frac{\tilde{k} - k}{k} \approx \frac{1}{24}\left\{\left[(k\Delta x)^2 \cos^4(\phi) + (k\Delta y)^2 \sin^4(\phi)\right] \sin^4(\theta) + (k\Delta z)^2 \cos^4(\theta) - (\omega\Delta t)^2\right\}, \tag{2.124}$$

where ϕ and θ denote the propagation direction of the wave in spherical coordinates. A plot of this phase error for a uniform discretization density is shown in Fig. 2.18. As with the 2D case, we see that we will always have some amount of numerical dispersion. However, this can be minimized by increasing the mesh density, with the phase error decreasing quadratically as a function of increasing mesh density.

2.10 Introduction to Absorbing Boundary Conditions

Up to this point, we have only discussed relatively simple kinds of boundary conditions. Namely, the Dirichlet and Neumann conditions. For electromagnetic problems, these boundary conditions are typically suitable to handle boundaries that occur at interfaces with perfect electric conductors (PECs) and perfect magnetic conductors (PMCs). These boundary conditions can typically be made to work in a suitable manner for analyzing *closed regions*; i.e., problems that have a natural boundary that no energy is able to "escape/pass" through. Many waveguide problems can be cast into a form where a closed region is an appropriate form of analysis.

However, for handling *open problems* where we are interested in analyzing the fields produced by a device in an unbounded region we need to find a suitable boundary condition to keep the size of the computational domain to a manageable volume. Typical examples of open region problems include analyzing the radiated fields produced by an antenna or determining the scattered fields produced by the interaction of an incident wave with some kind of scatterer (e.g., a radar target). The goal of a boundary condition to terminate an open problem is for it to allow a propagating field to "pass through it" without producing any reflected fields that will propagate back into the computational domain and corrupt the true solution to the problem. This kind of boundary condition is typically referred to as an *absorbing boundary condition (ABC)* since it "absorbs" the wave that is incident upon it. There are many different kinds of ABCs that have been formulated over the years, each with their unique advantages and disadvantages. We will discuss a few of the most popular ABCs in this course. From a terminology perspective, we will refer to methods that utilize a mathematical boundary condition to achieve their absorbing effect an ABC. We will briefly discuss the other common approach at the end of this section.

2.10.1 ABC – 1D Case

We will begin by considering the mathematical boundary condition ABC for a 1D case. We will consider the 1D case first due to its simplicity and to gain insight into this kind of mathematical boundary condition. However, we will quickly find that the excellent performance of the 1D ABC is a special case due to finite difference methods often being able to be optimized for "ideal" performance in 1D situations (this is similar to what we saw with the numerical dispersion canceling exactly in the 1D FDTD, but not in the 2D FDTD).

We will now derive the 1D ABC for the situation illustrated in Fig. 2.19. For this scenario, we have an unbounded region that we need to terminate into a finite sized region to be able to perform a numerical analysis. We introduce this artificial terminating "surface" some distance away from our region of actual interest that contains whatever inhomogeneities and sources constitute the problem we are trying to solve. Although unimportant for a 1D case, the goal in a more practical analysis is to have our artificial terminating surface far enough away from the region of actual interest that all of the fields have become approximately like plane waves.

If we assume we are working with a z-polarized electric field propagating in the $+x$-direction, then far away from any inhomogeneities we can assume that the field takes on the

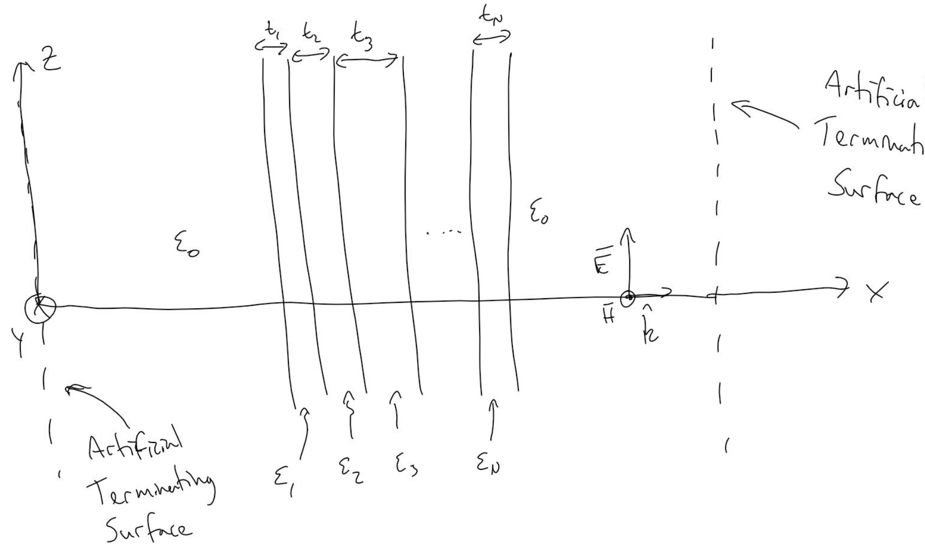


Figure 2.19: Illustration of the need for ABCs in the analysis of a 1D scattering problem.

form of a simple plane wave in the frequency domain like

$$E_z(x) = E_0 e^{-jkx}, \tag{2.125}$$

where E_0 is an arbitrary amplitude for the wave and k is the wavenumber. The ABC can be derived by taking the derivative of this plane wave with respect to x to get

$$\partial_x E_z = -jkE_0 e^{-jkx} = -j\frac{\omega}{c}E_z. \tag{2.126}$$

This gives us a relationship between the normal derivative of the field and the value of the field itself. This can be used as a boundary condition, and is often referred to as a *Robin boundary condition* or as a *boundary condition of the third kind*. The more general form is to have the field and its normal derivative equal to some other quantity, the case shown here is for a homogeneous Robin boundary condition since this “other” quantity is 0. We can now easily convert (2.126) into the time domain to get that

$$\boxed{\partial_x E_z(x,t) = -c^{-1}\partial_t E_z(x,t),} \tag{2.127}$$

which is our 1D ABC. Note that this ABC was derived for a wave propagating in the $+x$-direction, and so is only relevant for terminating our model at locations at one end of our model. The other end of the model can be terminated with a similar ABC that is derived for an electric field propagating in the $-x$-direction.

We can go about discretizing (2.127) in a few different ways. The simplest way is to use backward differencing for the spatial derivative and forward differencing for the temporal derivative at the edge of the model. Using these choices, leads to the time-stepping formula

$$E_z^{n+1}(I) = E_z^n(I) - \frac{c\Delta t}{\Delta x}[E_z^n(I) - E_z^n(I-1)], \tag{2.128}$$

where I is the maximum index along the x-grid. When implementing this kind of boundary condition it is also important to check that it will not unintentionally lead to numerical instability. A stability analysis can be performed for this equation to find that the stability condition is $\Delta t \leq \Delta x/c$, which matches the stability condition for a 1D finite difference discretization of the wave equation. Although this time-stepping formula is stable, the use of backward and forward differences does result in it only being first-order accurate.

To achieve a higher accuracy, it is necessary to devise a way to use central difference approximations to the derivatives. This can be done if we apply the central differencing formulas around the points $x = (I - 1/2)\Delta x$ and $t = (n + 1/2)\Delta t$. Doing this, (2.127) becomes

$$\frac{E_z^{n+1/2}(I) - E_z^{n+1/2}(I-1)}{\Delta x} = -\frac{1}{c}\frac{E_z^{n+1}(I-1/2) - E_z^{n}(I-1/2)}{\Delta t}. \tag{2.129}$$

We can follow the standard finite difference practice of computing values at "half-grid points" that we don't actually store fields at by using averages of nearby values. Doing this, we get

$$E_z^{n+1}(I) = E_z^{n}(I-1) + \frac{\Delta x - c\Delta t}{\Delta x + c\Delta t}[E_z^{n}(I) - E_z^{n+1}(I-1)]. \tag{2.130}$$

This formula is unconditionally stable and is also second-order accurate. As a result, this is the common way for an ABC to be implemented for a 1D FDTD method.

2.10.2 ABC – 2D Case

A similar derivation process can be used to derive an ABC for 2D (or 3D) cases. However, the immediate problem we are faced with is that in the 2D case we do not explicitly know the direction the "plane wave" will be propagating in when it hits our ABC. Due to this imprecise knowledge, we are unable to derive an ABC that will perform equally well for all propagation directions. We will now consider the derivation of two ABCs for the 2D case.

First-Order ABC

To begin, we will assume we are dealing with a boundary along the y-axis of a two-dimensional domain. Our assumed plane wave expression for the field will take on the general form

$$E_z(x,y) = E_0 e^{-j(k_x x + k_y y)}, \tag{2.131}$$

where $k_x = k\cos\theta$, $k_y = k\sin\theta$, and θ is the propagation direction of the wave. Since the boundary is along the y-axis, the wave will nominally be passing through it in the x-direction so we take the derivative with respect to x in the same way as we did for the 1D case. This gives us

$$\partial_x E_z = -jk_x E_z(x,y) = -jk\cos\theta E_z(x,y). \tag{2.132}$$

We can design this ABC to perfectly absorb a plane wave coming from only a single direction specified by θ. In general, actual fields can be viewed as being produced by a superposition of many plane waves propagating in different directions. Hence, we often will not have much ability/success in trying to optimize the ABC for a particular plane wave direction. Instead, we can just assume the plane wave approaches the boundary "head-on" so that $\theta = 0$. For this case, our approximate implementation of the ABC will become

$$\partial_x E_z \approx -jkE_z. \tag{2.133}$$

This is typically referred to as a *first-order ABC*.

To estimate the reflections that will occur due to our approximate termination of the computational model, we can compute the reflection coefficient as

$$\Gamma = \frac{Z_{W2} - Z_{W1}}{Z_{W2} + Z_{W1}}, \tag{2.134}$$

where Z_{W1} is the wave impedance of the wave inside the computational domain (which depends upon the propagation direction) and Z_{W2} is the effective wave impedance of our ABC implementation. The 2D simulation we are considering here corresponds to a perpendicular polarization case, for which we know that the wave impedance is

$$Z_{W1} = \frac{-E_z}{H_y} = \frac{\eta_1}{\cos\theta} \tag{2.135}$$

where $\eta_1 = \sqrt{\mu_1/\epsilon_1}$ is the intrinsic impedance of the computational domain near the ABC. For our Z_{W2} we can use Maxwell's equations to rewrite our ABC as

$$\begin{aligned} -j\frac{\omega}{c}E_z &= \partial_x E_z \\ &= j\omega\mu H_y. \end{aligned} \tag{2.136}$$

This can be rearranged to find that

$$Z_{W2} = \frac{-E_z}{H_y} = \eta_1. \tag{2.137}$$

If we had instead assumed a different value for θ in the implementation of our ABC, we would end up with a slightly modified form for Z_{W2}. However, for the case shown in (2.137), we can plug in for our two wave impedances to see that the reflection coefficient becomes

$$\boxed{\Gamma = \frac{\cos\theta - 1}{\cos\theta + 1}.} \tag{2.138}$$

This clearly becomes 0 when the wave is actually incident with $\theta = 0$, but will in general be non-zero. We also see that for *grazing angles* (i.e., large θ) this reflection coefficient can become rather large, making the ABC ineffective. Due to this, it is necessary to place the ABC some distance away from any inhomogeneities in the model so that the waves have enough time to begin spreading out so that there is a better likelihood of the wave approximately looking like a plane wave incident from the $\theta = 0$ direction.

Engquist-Majda ABC

As suggested by the name “first-order ABC”, it is possible for us to derive improved ABCs. In general, improving the ABC involves deriving a more complicated formula and then discretizing additional terms compared to a first-order ABC. However, the improvements in accuracy can often be worth this additional theoretical and computational work.

To see one way to improve the accuracy of the ABC we can rewrite (2.132) as

$$\partial_x E_z(x,y) = -jk_x E_z(x,y) = -j\sqrt{k^2 - k_y^2}E_z(x,y) = -jk\sqrt{1-\left(\frac{k_y}{k}\right)^2}E_z(x,y). \tag{2.139}$$

Based off of our assumptions for the waves reaching this boundary to be at angles with small θ, we can conclude that k_y/k should be relatively small as well. If we expand the square root in (2.139) using a Taylor series and only keep the first term, we would end up with the result we had in (2.133), which is the origin of the name “first-order ABC” for this particular implementation. If we instead keep the first two terms of the Taylor series we will get

$$\partial_x E_z(x,y) \approx -jkE_z(x,y) + \frac{j}{2k}k_y^2 E_z(x,y). \tag{2.140}$$

We can recognize that $\partial_y^2 E_z = -k_y^2 E_z$, so that we can rewrite this as

$$\partial_x E_z(x,y) \approx -jkE_z(x,y) - \frac{j}{2k}\partial_y^2 E_z(x,y). \tag{2.141}$$

This is a *second-order ABC*. The reflection coefficient for this ABC can also be computed, which is

$$\Gamma = \frac{\cos\theta + \frac{1}{2}\sin^2\theta - 1}{\cos\theta - \frac{1}{2}\sin^2\theta + 1}. \tag{2.142}$$

This reflection coefficient performs much better than that of the first-order ABC as a function of θ, with the comparison shown in Fig. 2.20.

We can convert (2.141) into the time domain by trading the wavenumber for ω/c and then replacing $j\omega$ with a time derivative. Doing this, we can eventually arrive at

$$\partial_t\partial_x E_z \approx -\frac{1}{c}\partial_t^2 E_z + \frac{c}{2}\partial_y^2 E_z, \tag{2.143}$$

which is often called the *Engquist-Majda absorbing boundary condition.*

We can discretize the different terms in (2.143) in the following way:

$$\begin{aligned}\partial_t\partial_x E_z &\approx \partial_t \frac{E_z^{n+1/2}(I,j) - E_z^{n+1/2}(I-1,j)}{\Delta x} \\ &\approx \frac{[E_z^{n+1}(I,j) - E_z^{n+1}(I-1,j)] - [E_z^{n}(I,j) - E_z^{n}(I-1,j)]}{\Delta x\,\Delta t},\end{aligned} \tag{2.144}$$

$$\partial_t^2 E_z \approx \frac{E_z^{n+1}(I,j) - 2E_z^{n}(I,j) + E_z^{n-1}(I,j)}{(\Delta t)^2}, \tag{2.145}$$

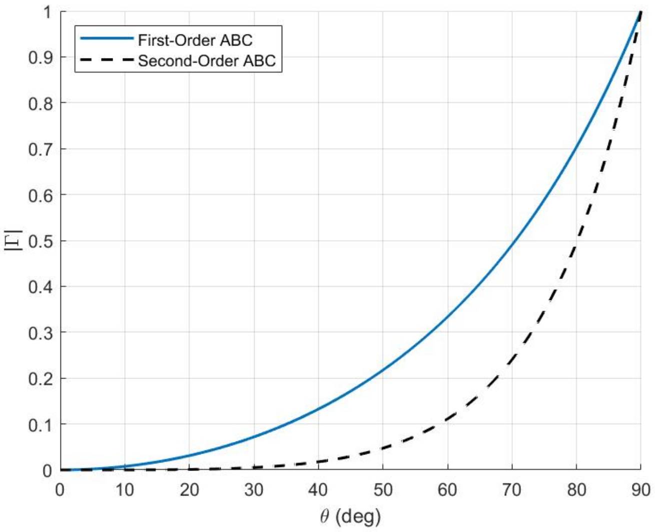


Figure 2.20: Comparison of the reflection coefficients produced by the first- and second-order ABCs.

$$\partial_y^2 E_z \approx \frac{E_z^n(I,j+1) - 2E_z^n(I,j) + E_z^n(I,j-1)}{(\Delta y)^2}. \tag{2.146}$$

These can then be used to develop a time-stepping formula of

$$\begin{aligned} E_z^{n+1}(I,j) = \left[\frac{1}{\Delta x \Delta t} + \frac{1}{c(\Delta t)^2}\right]^{-1} \Bigg\{ &\frac{1}{\Delta x \Delta t}\Big[E_z^{n+1}(I-1,j) - E_z^n(I-1,j)\Big] \\ &+ \left[\frac{1}{\Delta x \Delta t} + \frac{2}{c(\Delta t)^2} - \frac{c}{(\Delta y)^2}\right] E_z^n(I,j) - \frac{1}{c(\Delta t)^2} E_z^{n-1}(I,j) \\ &+ \frac{c}{2(\Delta y)^2}\Big[E_z^n(I,j-1) + E_z^n(I,j+1)\Big]\Bigg\}. \end{aligned} \tag{2.147}$$

This is obviously substantially more complicated than the first-order ABC. However, it is not so much more complicated that it cannot be readily implemented. This trend will continue if one were to try and derive even higher-order ABCs. As a result, pursuing higher and higher orders of ABC is not typically the best strategy to improve the performance of a simulation. In the next class, we will learn about *perfectly matched layers (PMLs)*, which are an alternative approach to terminate an open computational domain that involves the use of fictitious absorbing materials to enclose the simulation domain. As we will see, one benefit of the PML approach is that its performance can be systematically improved without reformulating the entire method, one simply needs to use more absorber. This does come at the price of increased computational cost, but this is often a fair trade-off compared to deriving and implementing increasingly higher-order ABCs.

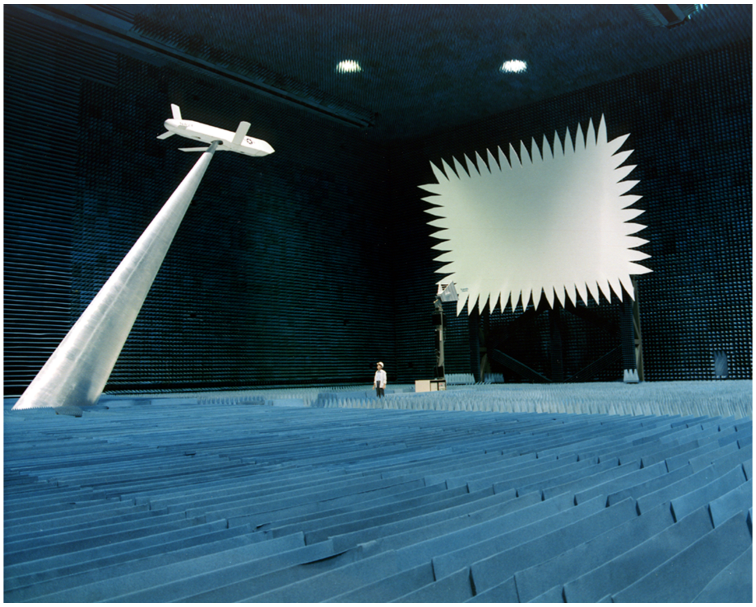


Figure 2.21: Example of a compact range (a special kind of anechoic chamber) at Point Mugu Naval Base in California (image from NSI-MI Technologies).

2.11 Perfectly Matched Layers

Previously, we discussed how to use ABCs to terminate the computational region considered when analyzing an open region electromagnetic problem. We saw that improving the performance of the ABC required developing a more sophisticated mathematical model, determining a suitable finite difference discretization strategy for the resulting equation, and then implementing a new set of code for the updated equation. This process is time-consuming and labor-intensive, which are significant drawbacks for this kind of method. Further issues can also occur for ABCs, such as how to formulate them for inhomogeneous media, lossy materials, etc.

We will now consider an alternative approach to terminate the computational domain known as a *perfectly matched layer (PML)*. The basic idea of a PML is to create an *ideal* anechoic chamber in our simulation. Anechoic chambers are common measurement facilities that are useful for measuring the performance of antennas and similar devices. These chambers are designed to replicate the general effect of a completely open region by covering the walls, ceiling, and floor with specially-designed broadband absorbers meant to minimize any reflections of electromagnetic fields that are incident on the absorbers (see Fig. 2.21). They are popular because they allow these kinds of measurements to be performed in much smaller spaces (e.g., a room inside a building) compared to large outdoor ranges (which also only approximate an open region).

Actually modeling an anechoic chamber with an FDTD code would be extremely computationally intensive due to the size of the absorbers, their frequency-dependent properties, and complicated shapes. However, since we are simply performing a simulation we have much more freedom in designing our absorbers than one has in “real life”. As a result, we can use fictitious materials that do not exist in reality but are designed to provide significant absorption of incident waves, produce minimal (ideally no) reflections, and are reasonably computationally efficient to model. A PML is exactly this kind of fictitious absorber. It can be placed around a simulation domain and then terminated on one side with a PEC boundary condition to enclose the entire simulation into a finite-sized region. If the absorber is designed properly, the amount of energy that "leaks" back into our simulation region of interest can be kept extremely small so that its influence is largely negligible.

Over the years, many different derivation approaches have been devised to go about formulating a PML. The initial formulation of the PML proposed in [8] used a somewhat non-intuitive development and involved the use of non-physical "split" fields. A slightly more intuitive derivation was eventually determined that utilized a form of Maxwell's equations in a specially stretched coordinate system [9]. However, this still involved non-physical split fields, so it is not truly "intuitive" either. Eventually, it was determined that the same behavior as these earlier PML formulations could be achieved assuming a special kind of anisotropic absorber [10, 11]. This formulation also suggested a different discretization approach that leads to new sets of time-stepping equations compared to the split field approaches developed previously. Although an anisotropic absorber is slightly more intuitive than coordinate stretching, it still requires the use of non-standard definitions for certain quantities (e.g., the magnetic and electric flux densities) to arrive at simple formulas to discretize the equations using the FDTD technique. This may seem slightly concerning at first, but it does not present any significant issue since the fields within the PML region are not of interest so long as they have minimal reflections back into the simulation region of actual interest.

We will now review the basics of the coordinate stretching and anisotropic absorber approaches to implementing PMLs. As alluded to previously, these methods lead to different sets of time-stepping equations and so differ in implementation. However, the basic details of the PML stay the same between the two methods. That is, the PML theoretically acts as a special material medium that does not reflect incoming plane waves regardless of angle of incidence, frequency, or polarization. In reality, the specific approximations made in discretizing PMLs will cause their numerical implementation to differ from their theoretical properties. Yet, the PMLs still can be designed to provide exceptionally good performance if done cleverly, so these minor numerical imperfections are often of little concern in practice.

2.11.1 Stretched Coordinate PML

General Theory

We will begin with the stretched coordinate PML. For this approach, we define three different stretching functions that each stretch one of the coordinate axes of our Cartesian system. We further will assume that each stretching function works only along a particular axis, so that they are of the form $s_x(x)$, $s_y(y)$, $s_z(z)$. At this point these functions are relatively arbitrary. However, the basic idea is that within the actual simulation domain we will have the stretching factors all equal 1 so that there is no stretching, but in the PML region the stretching degrees of freedom will be used to achieve the desired absorbing behavior. Now, within this stretched system the source-free Maxwell's equations are modified to be

$$\nabla_s \times \mathbf{E} = -j\omega\mu\mathbf{H} \tag{2.148}$$

$$\nabla_s \times \mathbf{H} = j\omega\epsilon\mathbf{E} \tag{2.149}$$

$$\nabla_s \cdot (\epsilon\mathbf{E}) = 0 \tag{2.150}$$

$$\nabla_s \cdot (\mu\mathbf{H}) = 0, \tag{2.151}$$

where

$$\nabla_s = \hat{x}s_x^{-1}\partial_x + \hat{y}s_y^{-1}\partial_y + \hat{z}s_z^{-1}\partial_z. \tag{2.152}$$

To gain insight into how the stretching factors can be used, it will help us to examine the characteristics of plane waves within a homogeneous medium using our stretched form of Maxwell's equations. Our plane wave solutions will still be in the form of

$$\mathbf{E} = \mathbf{E}_0 e^{-j(k_x x + k_y y + k_z z)}, \tag{2.153}$$

with a similar definition for $\mathbf{H}$. These plane waves can be substituted into the stretched form of Maxwell's equations. Evaluating the derivatives and consolidating terms allows us to see that for a plane wave Maxwell's equations become

$$\mathbf{k}_s \times \mathbf{E} = \omega\mu\mathbf{H} \tag{2.154}$$

$$\mathbf{k}_s \times \mathbf{H} = -\omega\epsilon\mathbf{E} \tag{2.155}$$

$$\mathbf{k}_s \cdot \mathbf{E} = 0 \tag{2.156}$$

$$\mathbf{k}_s \cdot \mathbf{H} = 0, \tag{2.157}$$

where

$$\mathbf{k}_s = \hat{x}\frac{k_x}{s_x} + \hat{y}\frac{k_y}{s_y} + \hat{z}\frac{k_z}{s_z}. \tag{2.158}$$

We can now use (2.154) to (2.157) to study the dispersion relation of the plane wave within the stretched coordinate system. We do this by forming a "wave equation" using (2.154) to (2.157) to get

$$\mathbf{k}_s \times \mathbf{k}_s \times \mathbf{E} = -\omega^2\mu\epsilon\mathbf{E}. \tag{2.159}$$

We can use the standard vector algebraic identity that $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = (\mathbf{a}\cdot\mathbf{c})\mathbf{b} - (\mathbf{a}\cdot\mathbf{b})\mathbf{c}$ and the fact that $\mathbf{k}_s \cdot \mathbf{E} = 0$ in a source-free region to rewrite this as

$$(\mathbf{k}_s \cdot \mathbf{k}_s)\mathbf{E} = \omega^2\mu\epsilon\mathbf{E} = k^2\mathbf{E}. \tag{2.160}$$

We quickly see that

$$\left(\frac{k_x}{s_x}\right)^2 + \left(\frac{k_y}{s_y}\right)^2 + \left(\frac{k_z}{s_z}\right)^2 = k^2, \tag{2.161}$$

which has solution

$$k_x = ks_x \sin\theta\cos\phi \tag{2.162}$$

$$k_y = ks_y \sin\theta\sin\phi \tag{2.163}$$

$$k_z = ks_z \cos\theta. \tag{2.164}$$

These closely mirror the standard solution for the dispersion relation of plane waves in a homogeneous medium, with the added degree of freedom provided by the stretching factors. The idea of the PML is that if these stretching factors are made into appropriately-defined complex-valued numbers they can produce attenuation of the wave as it propagates. Importantly, we can also check and see that the wave impedance in the stretched coordinates can be found to be

$$Z_W = \frac{|\mathbf{E}|}{|\mathbf{H}|} = \frac{|\mathbf{k}_s|}{\omega\epsilon} = \sqrt{\frac{\mu}{\epsilon}} = \eta. \tag{2.165}$$

That is, it is independent of the stretching. This gives us one of our first signs that if we are careful we may be able to define our stretching parameters in such a way that a wave propagating in a region with material properties μ and ϵ and no stretching won't produce a reflection if it reaches a region where there is stretching.

To see how to implement this, consider the reflection of an oblique incidence plane wave from an interface between two regions with homogeneous stretching that meet at the $z = 0$ plane. It is possible to work out all the algebraic equations to enforce the continuity of the tangential components of the electric and magnetic fields so that we can determine the reflection coefficient for this scenario. We won't go into these details here, but will instead note that the final result for TE and TM polarizations are

$$R_{TE} = \frac{(k_{1z}/s_{1z})\mu_2 - (k_{2z}/s_{2z})\mu_1}{(k_{1z}/s_{1z})\mu_2 + (k_{2z}/s_{2z})\mu_1}, \tag{2.166}$$

$$R_{TM} = \frac{(k_{1z}/s_{1z})\epsilon_2 - (k_{2z}/s_{2z})\epsilon_1}{(k_{1z}/s_{1z})\epsilon_2 + (k_{2z}/s_{2z})\epsilon_1}, \tag{2.167}$$

and that $k_{1x} = k_{2x}$ and $k_{1y} = k_{2y}$ due to the phase matching condition at the boundary. If we take a closer look at the numerator of (2.166), we can rewrite this using our dispersion relation to have

$$(k_{1z}/s_{1z})\mu_2 - (k_{2z}/s_{2z})\mu_1 = \mu_2\sqrt{k^2 - \left(\frac{k_{1x}}{s_{1x}}\right)^2 - \left(\frac{k_{1y}}{s_{1y}}\right)^2} - \mu_1\sqrt{k^2 - \left(\frac{k_{2x}}{s_{2x}}\right)^2 - \left(\frac{k_{2y}}{s_{2y}}\right)^2}. \tag{2.168}$$

This will equal 0 if we make $s_{1x} = s_{2x}$, $s_{1y} = s_{2y}$, and $\mu_1 = \mu_2$. If we also make $\epsilon_1 = \epsilon_2$, then the numerator of (2.167) will also equal 0. It is important to note that these results will hold for all θ and ϕ, as well as for all s_{1z} and s_{2z}. Hence, we can choose s_{1z} and s_{2z} arbitrarily without producing any reflections from our boundary.

We can exploit this degree of freedom to make waves propagating in the z-direction in one of our regions attenuate. For instance, if we say that Region 1 is our region of interest in the simulation then we can set $s_{1x} = s_{1y} = s_{1z} = 1$ so that there is no stretching and we can produce unperturbed solutions to Maxwell's equations in this region. We can then choose to set $s_{2x} = s_{2y} = 1$ and $s_{2z} = s' - js''$ where s' and s'' are real-valued positive numbers. Then the propagation constant in the z-direction in Region 2 (our PML) will become

$$k_{2z} = k_2(s' - js'')\cos\theta, \tag{2.169}$$

and so we will have attenuation in the z-direction, as desired. As mentioned previously, to keep our simulation problem finite-sized we will need to terminate the back of the PML with a boundary condition. Typically, this is taken to be a PEC plane for simplicity, although other options have also been studied. If the total thickness of the PML region is L and it is terminated in a PEC plane then we can quickly find that the total reflection produced from our terminated PML will be

$$|R(\theta)| = \exp\left[-2k_2\cos\theta\int_0^L s''(z)dz\right]. \tag{2.170}$$

Obviously, if the plane wave reaches the PML at a near-grazing angle the reflection will reach toward its maximum values. Hence, just as with the ABCs, we still need to be careful and place our PML some distance away from our simulation region of interest so that the waves reaching the PML are more like a normal incident plane wave. Although this characteristic is similar to an ABC, the important distinction of the PML is that we can *systematically* improve its performance in simple ways without reformulating our approach or changing our numerical implementation. For instance, if we need smaller reflections we can increase the thickness of the PML or increase the attenuation by increasing s''.

From our theoretical formulation, it would seem that we could use a very large s'' and a thin PML to get very low reflections. In reality, the finite difference approximations that we will use to discretize the PML will cause our numerical behavior to deviate from this ideal theoretical behavior. Hence, reflections at the interface of the PML and the simulation region of interest may occur. This numerical artifact will be exacerbated if we have a very large s'', since it will cause more of a “jump” change in properties compared to the non-stretched simulation region. To minimize these kinds of numerical artifacts, it is common

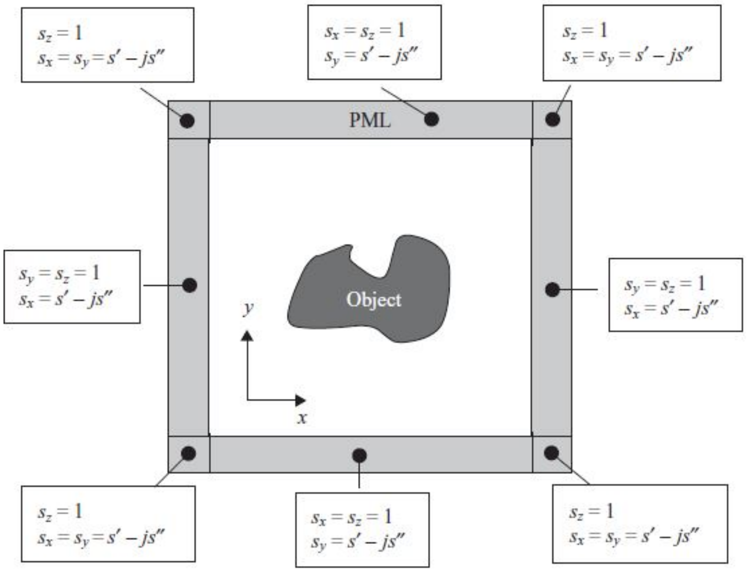


Figure 2.22: Layout of the PML parameters to achieve the desired behavior along all directions surrounding the computational domain of interest (image from [5]).

to implement a PML with spatially-varying values of $s''(z)$. By starting with a small $s''(z)$ and gradually increasing it, we can minimize reflections due to numerical artifacts and still produce an overall large amount of attenuation in the end.

We can generalize our results to design PMLs to completely enclose our simulation region of interest, as shown in Fig. 2.22. The main extra detail to consider is what to do in the "corners" and "edges" where we have PMLs attached to different surfaces overlapping. It turns out, that the desired performance of the PML will be maintained if we stretch multiple coordinates simultaneously.

Finite Difference Implementation

We are now ready to determine how to implement the PML in the FDTD method. In the previous section, we performed our analysis in the frequency domain to gain insight into how the coordinate stretching would affect the electromagnetic fields. If we attempt to directly transform the PML equations into the time domain we can quickly run into some complications on how to effectively deal with certain terms. The approach taken in the original implementations of the PML was to utilize "non-physical" *split fields.* These split fields allow us to separate Maxwell's curl equations into split components so that the stretching factors do not "interact" with each other and complicate the transformation to the time domain and corresponding finite difference discretization.

To see this, we begin by noting that in the stretched coordinates we have

$$\nabla_s \times \mathbf{E} = s_x^{-1}\partial_x(\hat{x} \times \mathbf{E}) + s_y^{-1}\partial_y(\hat{y} \times \mathbf{E}) + s_z^{-1}\partial_z(\hat{z} \times \mathbf{E}). \tag{2.171}$$

We can then write Faraday's law $\nabla_s \times \mathbf{E} = -j\omega\mu\mathbf{H}$ into three *vector equations* by decomposing the magnetic field into three vector components as $\mathbf{H} = \mathbf{H}_{sx} + \mathbf{H}_{sy} + \mathbf{H}_{sz}$. Our three vector equations are then

$$s_x^{-1}\partial_x(\hat{x}\times\mathbf{E}) = -j\omega\mu\mathbf{H}_{sx} \tag{2.172}$$

$$s_y^{-1}\partial_y(\hat{y}\times\mathbf{E}) = -j\omega\mu\mathbf{H}_{sy} \tag{2.173}$$

$$s_z^{-1}\partial_z(\hat{z}\times\mathbf{E}) = -j\omega\mu\mathbf{H}_{sz}. \tag{2.174}$$

To have a simple time domain implementation, we need to choose a relatively simple form for s_x, s_y, and s_z. In particular, we can have them use fictitious conductivities that are frequency independent so that they look like

$$s_x = 1 - j\frac{\sigma_x}{\omega\epsilon}, \tag{2.175}$$

with similar forms for s_y and s_z. We can now convert (2.172) to (2.174) into the time domain easily to get

$$\partial_x(\hat{x}\times\mathbf{E}) = -\mu\partial_t\mathbf{H}_{sx} - \sigma_x\mu\epsilon^{-1}\mathbf{H}_{sx} \tag{2.176}$$

$$\partial_y(\hat{y}\times\mathbf{E}) = -\mu\partial_t\mathbf{H}_{sy} - \sigma_y\mu\epsilon^{-1}\mathbf{H}_{sy} \tag{2.177}$$

$$\partial_z(\hat{z}\times\mathbf{E}) = -\mu\partial_t\mathbf{H}_{sz} - \sigma_z\mu\epsilon^{-1}\mathbf{H}_{sz}. \tag{2.178}$$

Note that we are only able to get this simple of equations in the time domain due to our use of the split fields. A similar splitting process can also be used for the electric field so that $\mathbf{E} = \mathbf{E}_{sx} + \mathbf{E}_{sy} + \mathbf{E}_{sz}$ and Ampere's law in the time domain becomes

$$\partial_x(\hat{x}\times\mathbf{H}) = \epsilon\partial_t\mathbf{E}_{sx} + \sigma_x\mathbf{E}_{sx} \tag{2.179}$$

$$\partial_y(\hat{y}\times\mathbf{H}) = \epsilon\partial_t\mathbf{E}_{sy} + \sigma_y\mathbf{E}_{sy} \tag{2.180}$$

$$\partial_z(\hat{z}\times\mathbf{H}) = \epsilon\partial_t\mathbf{E}_{sz} + \sigma_z\mathbf{E}_{sz}. \tag{2.181}$$

Yee's FDTD scheme can now be applied to discretize (2.176) to (2.181). Doing this for the full 3D case is somewhat tedious, so we will consider only a 2D problem here. In particular, if we consider a TM_z problem we will have that $\mathbf{E} = \hat{z}E_z$ and $\mathbf{H} = \hat{x}H_x + \hat{y}H_y$. If we utilize these in (2.178) we can find that $\mathbf{H}_{sz} = 0$ (Note: $\sigma_z = 0$ here due to the required invariance in z for our 2D analysis). Then, we can see that the rest of Faraday's law in (2.176) and (2.177) become

$$\partial_x E_z = \mu\partial_t H_y + \sigma_x\mu\epsilon^{-1}H_y \tag{2.182}$$

$$\partial_y E_z = -\mu\partial_t H_x - \sigma_y\mu\epsilon^{-1}H_x. \tag{2.183}$$

Due to the simplicity of the 2D case, it seems that the “splitting” of the fields has vanished in a way. However, this does not happen for the Ampere’s law equations, and so we see that the splitting is still necessary and does not completely vanish. In particular, we can see from (2.181) that $\mathbf{E}_{sz} = 0$ due to the uniformity along the z-axis of the 2D problem, but that we will have $\mathbf{E}_{sx} = \hat{z}E_{sx,z}$ and $\mathbf{E}_{sy} = \hat{z}E_{sy,z}$ from (2.179) and (2.180), respectively. We can then find that the explicit forms for (2.179) and (2.180) simplify to

$$\partial_x H_y = \epsilon\partial_t E_{sx,z} + \sigma_x E_{sx,z} \tag{2.184}$$

$$\partial_y H_x = -\epsilon\partial_t E_{sy,z} - \sigma_y E_{sy,z}. \tag{2.185}$$

The resulting time stepping equations for this system becomes:

$$\begin{aligned} H_x^{n+1/2}(i,j+1/2) = a_y(i,j+1/2)\Bigg\{ & b_y(i,j+1/2)H_x^{n-1/2}(i,j+1/2) \\ & - \frac{\epsilon}{\mu\Delta y}\Big[E_z^n(i,j+1) - E_z^n(i,j)\Big]\Bigg\}, \end{aligned} \tag{2.186}$$

$$\begin{aligned} H_y^{n+1/2}(i+1/2,j) = a_x(i+1/2,j)\Bigg\{ & b_x(i+1/2,j)H_y^{n-1/2}(i+1/2,j) \\ & + \frac{\epsilon}{\mu\Delta x}\Big[E_z^n(i+1,j) - E_z^n(i,j)\Big]\Bigg\}, \end{aligned} \tag{2.187}$$

$$E_{sx,z}^{n+1}(i,j) = a_x(i,j)\Bigg\{b_x(i,j)E_{sx,z}^n(i,j) + \frac{1}{\Delta x}\Big[H_y^{n+1/2}(i+1/2,j) - H_y^{n+1/2}(i-1/2,j)\Big]\Bigg\}, \tag{2.188}$$

$$E_{sy,z}^{n+1}(i,j) = a_y(i,j)\Bigg\{b_y(i,j)E_{sx,z}^n(i,j) - \frac{1}{\Delta y}\Big[H_x^{n+1/2}(i,j+1/2) - H_x^{n+1/2}(i,j-1/2)\Big]\Bigg\}, \tag{2.189}$$

with

$$a_{x(y)} = \left[\frac{\epsilon}{\Delta t} + \frac{\sigma_{x(y)}}{2}\right]^{-1} \tag{2.190}$$

$$b_{x(y)} = \left[\frac{\epsilon}{\Delta t} - \frac{\sigma_{x(y)}}{2}\right]. \tag{2.191}$$

2.11.2 Anisotropic Absorber PML

As mentioned previously, it is also possible to derive a PML in terms of an uniaxial absorber. An intuitive derivation of the form of uniaxial medium can be found in [10, 11] for the simple case of only considering a single plane boundary. Unfortunately, when this derivation needs to be extended to handle "corner" and "edge" regions where two PMLs overlap the derivation loses its physical intuition. Considering this difficulty, we will only focus on how the anisotropic absorber perspective can be derived from the coordinate stretching equations. We will then consider the process for discretizing the resulting equations in the FDTD method.

Now, our goal is to convert the stretched coordinate form of Maxwell's equations into something that looks like a standard set of Maxwell's equations. This can be done by defining a new set of fields in terms of the stretched ones as

$$\mathbf{E}^a = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \cdot \mathbf{E}^s, \tag{2.192}$$

where $\mathbf{E}^a$ is our new field (the a denotes this will be for our anisotropic absorber equations) and $\mathbf{E}^s$ is the field used in the stretched coordinate form of Maxwell's equations (the s denotes stretched coordinate equations). A similar definition also holds for $\mathbf{H}^a$ and $\mathbf{H}^s$. We now want to see how to convert an equation like

$$\nabla_s \times \mathbf{E}^s = -j\omega\mu \mathbf{H}^s \tag{2.193}$$

into a form that uses $\mathbf{E}^a$ and $\mathbf{H}^a$. To do this, we will need to see how to re-express $\nabla_s \times \mathbf{E}^s$. We can explicitly compute this to get

$$\begin{aligned}\nabla_s \times \mathbf{E}^s = \begin{bmatrix} s_y^{-1}\partial_y E_z^s - s_z^{-1}\partial_z E_y^s \\ s_z^{-1}\partial_z E_x^s - s_x^{-1}\partial_x E_z^s \\ s_x^{-1}\partial_x E_y^s - s_y^{-1}\partial_y E_x^s \end{bmatrix} &= \begin{bmatrix} (s_y s_z)^{-1}(\partial_y E_z^a - \partial_z E_y^a) \\ (s_z s_x)^{-1}(\partial_z E_x^a - \partial_x E_z^a) \\ (s_x s_y)^{-1}(\partial_x E_y^a - \partial_y E_x^a) \end{bmatrix} \\ &= \begin{bmatrix} (s_y s_z)^{-1} & 0 & 0 \\ 0 & (s_z s_x)^{-1} & 0 \\ 0 & 0 & (s_x s_y)^{-1} \end{bmatrix} \cdot \nabla \times \mathbf{E}^a. \end{aligned} \tag{2.194}$$

Hence, we can write (2.193) as

$$\begin{bmatrix} (s_y s_z)^{-1} & 0 & 0 \\ 0 & (s_z s_x)^{-1} & 0 \\ 0 & 0 & (s_x s_y)^{-1} \end{bmatrix} \cdot \nabla \times \mathbf{E}^a = -j\omega\mu \begin{bmatrix} (s_x)^{-1} & 0 & 0 \\ 0 & (s_y)^{-1} & 0 \\ 0 & 0 & (s_z)^{-1} \end{bmatrix} \cdot \mathbf{H}^a. \tag{2.195}$$

We can consolidate this as

$$\nabla \times \mathbf{E}^a = -j\omega\mu \overline{\mathbf{\Lambda}} \cdot \mathbf{H}^a, \tag{2.196}$$

where

$$\overline{\mathbf{\Lambda}} = \begin{bmatrix} \dfrac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \dfrac{s_z s_x}{s_y} & 0 \\ 0 & 0 & \dfrac{s_x s_y}{s_z} \end{bmatrix}. \tag{2.197}$$

We can readily see that (2.196) is Faraday's law for a uniaxial anisotropic medium. We can similarly arrive at

$$\nabla \times \mathbf{H}^a = j\omega\epsilon\overline{\mathbf{\Lambda}} \cdot \mathbf{E}^a \tag{2.198}$$

$$\nabla \cdot (\epsilon\overline{\mathbf{\Lambda}} \cdot \mathbf{E}^a) = 0 \tag{2.199}$$

$$\nabla \cdot (\mu\overline{\mathbf{\Lambda}} \cdot \mathbf{H}^a) = 0. \tag{2.200}$$

Finally, we arrive at the desired result; i.e., the stretched coordinate form of Maxwell's equations can be mapped to a regular form of Maxwell's equations with permittivity and permeability tensors of $\overline{\boldsymbol{\epsilon}} = \epsilon\overline{\mathbf{\Lambda}}$ and $\overline{\boldsymbol{\mu}} = \mu\overline{\mathbf{\Lambda}}$, respectively.

To achieve performance similar to the stretched coordinate PML, we will need to adopt similar definitions for the different stretching parameters used in the definition of $\overline{\mathbf{\Lambda}}$. In particular, recall that we will need to have each of the stretching parameters defined similar to

$$s_x = 1 - j\frac{\sigma_x}{\omega\epsilon}. \tag{2.201}$$

This is where the difficulty comes into play when considering how to convert these equations to the time domain for "corner" and "edge" regions where we simultaneously have multiple conductivities that are non-zero. This leads to a material property tensor that has a complex non-linear frequency dependence that is non-trivial to implement in the time domain.

However, it is possible to circumvent these issues by defining a set of auxiliary vectors that are *almost* equivalent to the electric and magnetic flux densities in the anisotropic material. We can then follow a two step updating process where these auxiliary vectors are computed as additional steps in our leapfrog time marching process.

To see how this is done, we will first define our auxiliary vectors $\mathbf{D}$ and $\mathbf{B}$ as

$$\mathbf{D} = \epsilon \begin{bmatrix} \frac{s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x}{s_y} & 0 \\ 0 & 0 & \frac{s_y}{s_z} \end{bmatrix} \cdot \mathbf{E}, \tag{2.202}$$

$$\mathbf{B} = \mu \begin{bmatrix} \frac{s_z}{s_x} & 0 & 0 \\ 0 & \frac{s_x}{s_y} & 0 \\ 0 & 0 & \frac{s_y}{s_z} \end{bmatrix} \cdot \mathbf{H}, \tag{2.203}$$

where we have dropped the superscript a for notational simplicity. The purpose of these definitions is to simplify Faraday's and Ampere's laws to be

$$\nabla \times \mathbf{E} = -j\omega \begin{bmatrix} s_y & 0 & 0 \\ 0 & s_z & 0 \\ 0 & 0 & s_x \end{bmatrix} \cdot \mathbf{B} \tag{2.204}$$

$$\nabla \times \mathbf{H} = j\omega \begin{bmatrix} s_y & 0 & 0 \\ 0 & s_z & 0 \\ 0 & 0 & s_x \end{bmatrix} \cdot \mathbf{D}. \tag{2.205}$$

Both of these equations involve only a single stretching parameter in each scalar equation, so it will be easy to transform them to the time domain. Likewise, we can rewrite (2.202) and (2.203) as

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \cdot \mathbf{D} = \epsilon \begin{bmatrix} s_z & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_y \end{bmatrix} \cdot \mathbf{E} \tag{2.206}$$

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \cdot \mathbf{B} = \mu \begin{bmatrix} s_z & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & s_y \end{bmatrix} \cdot \mathbf{H}, \tag{2.207}$$

so that each scalar equation can be easily converted to the time domain.

For example, we can convert the x-components of (2.204) and (2.205) into the time domain as

$$\partial_y E_z - \partial_z E_y = -\partial_t B_x - \frac{\sigma_y}{\epsilon} B_x \tag{2.208}$$

$$\partial_y H_z - \partial_z H_y = \partial_t D_x + \frac{\sigma_y}{\epsilon} D_x, \tag{2.209}$$

which can be converted into time-stepping equations using Yee's method. Note that for this implementation, $\mathbf{E}$ and $\mathbf{D}$ are discretized on the same grid and $\mathbf{H}$ and $\mathbf{B}$ are discretized on the same grid (i.e., the one dual to the grid used for $\mathbf{E}$ and $\mathbf{D}$). For example, the time-stepping equations for (2.208) is

$$\begin{aligned} B_x^{n+1/2}(i, j+1/2, k+1/2) = {} & a_y(i, j+1/2, k+1/2) \times \\ & \Big\{ b_y(i, j+1/2, k+1/2) B_x^{n-1/2}(i, j+1/2, k+1/2) \\ & - \frac{\epsilon}{\Delta y}\Big[E_z^n(i, j+1, k+1/2) - E_z^n(i, j, k+1/2)\Big] \\ & + \frac{\epsilon}{\Delta z}\Big[E_y^n(i, j+1/2, k+1) - E_y^n(i, j+1/2, k)\Big]\Big\}. \end{aligned} \tag{2.210}$$

We can also convert (2.206) and (2.207) to the time domain. The x-components of these equations become

$$\partial_t D_x + \frac{\sigma_x}{\epsilon} D_x = \epsilon \partial_t E_x + \sigma_z E_x \tag{2.211}$$

$$\partial_t B_x + \frac{\sigma_x}{\epsilon} B_x = \mu \partial_t H_x + \mu \frac{\sigma_z}{\epsilon} H_x. \tag{2.212}$$

The explicit time-stepping equations can be found easily; e.g., for (2.211) we get

$$E_x^{n+1}(i+1/2,j,k) = a_z\left[b_z E_x^n + \frac{1}{\epsilon a_x} D_x^{n+1} - \frac{b_x}{\epsilon} D_x^n\right], \tag{2.213}$$

where the arguments of all quantities on the right-hand side match that of the left-hand side. Note that to improve the accuracy, averages of values over adjacent time steps are used for the quantities in (2.211) and (2.212) that do not have a time derivative applied to them.

We see that our x-component equations have given us four equations to solve. We can follow a similar process to get 8 more equations between the y- and z-components. These can all be solved together in a leapfrog time-stepping process. In particular, the leapfrog time-stepping process now follows the following format.

1. Use (2.208) and the knowledge of $\mathbf{E}^n$ to compute $B_x^{n+1/2}$.
2. Use $B_x^{n+1/2}$ and previous values of B_x and H_x in (2.212) to compute $H_x^{n+1/2}$.
3. Use $H_x^{n+1/2}$ (and other vector components) in (2.209) to compute D_x^{n+1}.
4. Use D_x^{n+1} and previous values of D_x and E_x in (2.211) to compute E_x^{n+1}.

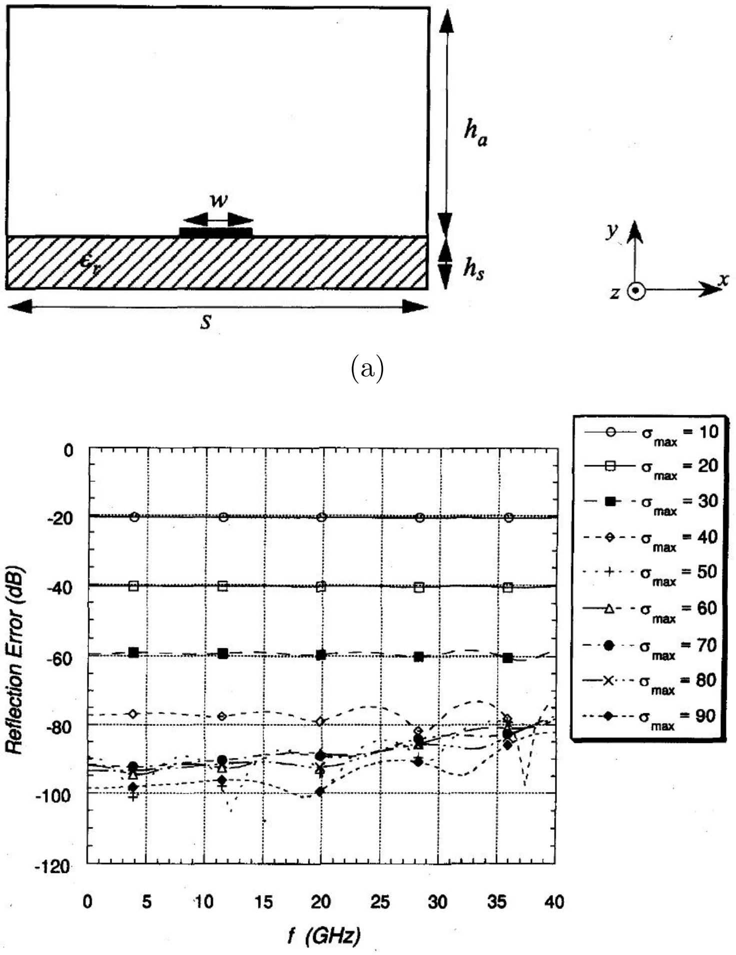
2.11.3 Some Concluding Remarks

As mentioned previously, one of the primary benefits of the PML over the ABCs is that the performance of the PML can be systematically improved in a simple manner by increasing the thickness or conductivity of the PML. To illustrate this, the reflection error for a ten-cell thick PML region with varying values of conductivities are plotted in Fig. 2.23. For this problem, the PML is being used as a termination for a microstrip transmission line. Further, to improve the smoothness of the transition from the simulation region to the PML, the conductivity profile is implemented as

$$\sigma(z) = \frac{\sigma_{\max}|z-z_0|^m}{L^m}, \tag{2.214}$$

where L is the thickness of the PML and z_0 is the reference point at the interface between the PML and the simulation region of interest. For the results in Fig. 2.23, a fourth-order polynomial is used (i.e., $m = 4$).

So far, we have largely focused on the advantages of using a PML, but it should be emphasized that they are not a panacea. It is important to remember that all of our derivations for their reflectionless properties were built on a propagating plane wave assumption. Whenever the fields approaching the PML region deviate from this assumption, reflections can occur and the PML will begin to have less ideal performance. One particular example that PMLs have difficulty in handling is evanescent waves. To address this, more complicated PML implementations have been devised with varying levels of success (see [5] for a short reference list).



(a)

(b)

Figure 2.23: Sample results demonstrating the efficacy of a PML. (a) Shielded microstrip geometry that is terminated at the ends with a PML and (b) reflection error for various conductivity profiles (images from [11]).

2.12 Modeling Dispersive Materials

Realistic materials always exhibit some amount of dispersion, i.e., their constitutive properties vary as a function of frequency. At a fundamental level, we typically associate this frequency variation with the inertia that causes the atoms and molecules that make up a material to not be able to respond instantaneously to changes in electric or magnetic fields. Accounting for this dispersion in the frequency domain is quite straightforward, we simply modify our material properties at each new frequency that we perform our analysis at.

However, the story is quite different in the time domain. When the material properties are no longer constant as a function of frequency, we find that the constitutive relations between the fields and fluxes take the form of a convolution; e.g.,

$$\mathbf{D}(\mathbf{r},\omega) = \epsilon(\mathbf{r},\omega)\mathbf{E}(\mathbf{r},\omega) \iff \mathbf{D}(\mathbf{r},t) = \epsilon(\mathbf{r},t) * \mathbf{E}(\mathbf{r},t), \tag{2.215}$$

where

$$\epsilon(\mathbf{r},t) * \mathbf{E}(\mathbf{r},t) = \int_0^t \epsilon(\mathbf{r},t-\tau)\mathbf{E}(\mathbf{r},\tau)d\tau \tag{2.216}$$

and we have assumed that both $\mathbf{E}$ and ϵ are 0 for all $t \leq 0$ (requiring $\epsilon(t) = 0$ for $t \leq 0$ is a consequence of the *causality* of the material; i.e., it cannot respond to an electric field before the electric field reaches it). We must be particularly careful with how we go about incorporating this convolution integral into our numerical discretization. In particular, if we do this in a naive manner we can significantly increase the computational cost of our simulation.

In the coming sections, we will discuss two different methods for efficiently handling the incorporation of dispersive materials into our FDTD analyses. In most respects, these two methods are able to achieve similar results; e.g., they have commensurate accuracy, require similar memory storage requirements, and involve similar numbers of floating point operations. In some cases, formulating one versus the other may be simpler. Likewise, the code implementation may have some advantages for one method over the other. However, in general, both methods are still popular. Before continuing on, we briefly mention that although we will only focus on handling dispersive materials here, similar methods can also be adapted to consider certain kinds of nonlinear materials [12].

2.12.1 Recursive Convolution

The first method we will consider is known as the *recursive convolution approach.* This method computes the results of the convolution integral directly. The computational cost of naively integrating this integral is often prohibitive, since it would require storing (and then using) all previous values of the electric field over and over again. To avoid this, a recursive computation is utilized that keeps the computational cost at a manageable level. Although it is not always possible, this kind of recursive computation for evaluating convolution integrals is a very common technique in numerical modeling.

To begin developing the recursive convolution process, recall from basic electromagnetic theory that we often can develop simple models for the frequency variation of the permittivity of a material in terms of the electric susceptibility χ_e of the material. Simple models for determining χ_e include the Debye or Drude-Lorentz-Sommerfeld models. For each different kind of model, a new recursive convolution procedure may need to be developed. Hence, it is better to think of the recursive convolution approach as a general strategy for efficiently discretizing a model, but is something that may need to be reformulated for each different scenario it is intended to be used in.

Now, one simple description of an electrically dispersive material is to have

$$\mathbf{D}(t) = \epsilon_\infty \mathbf{E}(t) + \epsilon_0 \int_0^t \tilde{\chi}_e(t-\tau)\mathbf{E}(\tau)d\tau, \tag{2.217}$$

where ϵ_∞ is the permittivity at “infinite frequency” (typically taken to be its optical value for many RF/microwave applications) and

$$\tilde{\chi}_e = (1 - \epsilon_\infty/\epsilon_0)\delta(t) + \chi_e(t). \tag{2.218}$$

Note that for notational simplicity, we are omitting the spatial argument of these functions. For most simple materials this does not cause an issue, however, there are some modern applications that have *non-local* permittivity functions that would need to be considered using a more detailed approach. Our goal is now to determine how to use this in the development of an FDTD set of equations. Considering that $\mathbf{D}$ will enter Ampere's law as

$$\nabla \times \mathbf{H} = \partial_t \mathbf{D} \tag{2.219}$$

for the source-free and zero conductivity case, we see that we should compute the time derivative of (2.217) to have it be in a useful form for inclusion in the Yee method.

Considering this, we need to determine how to discretize the right-hand side of

$$\partial_t D(t) = \epsilon_\infty \partial_t E(t) + \epsilon_0 \tilde{\chi}_e(t) * \partial_t E(t), \tag{2.220}$$

where we focus on only a single unspecified scalar component to simplify the notation. The first term on the right-hand side is simple and can be handled using a central difference approximation centered on the magnetic field's time step, i.e., at $t = (n+1/2)\Delta t$. We can write the partially time-discretized form of the second term on the right-hand side at the same time step as

$$\begin{aligned}\tilde{\chi}_e(t) * \partial_t E(t)|_{t=(n+1/2)\Delta t} \approx \int_0^{\Delta t/2} & \tilde{\chi}_e(\tau)\dot{E}(n\Delta t - \tau)d\tau \\ & + \sum_{k=0}^{n-1} \int_{(k+1/2)\Delta t}^{(k+3/2)\Delta t} \tilde{\chi}_e(\tau)\dot{E}(n\Delta t - \tau)d\tau, \end{aligned} \tag{2.221}$$

where we denote the time derivative on the right-hand side with an over-dot notation for simplicity. The first term on the right-hand side of (2.221) has to be separated from the rest of the integration because it only covers half of a time step due to the offset nature of the time grids for E and H. At this point, we can utilize some approximations to simplify the evaluation of these various integrals. The simplest approximation is to assume that the field quantities and their time derivatives take on a constant value over the entire length of a time step. This then allows $\dot{E}$ to be pulled outside of the time integrations and approximated using a central difference scheme. Higher-order approximations, such as assuming that the field quantities take on piecewise linear variation, are also possible but require reformulating the FDTD update equations for each new approach developed [13]. We will focus only on this simplest approximation (constant field values) to illustrate the basic idea of the recursive convolution method.

Now, utilizing this approximation and using a central difference for $\dot{E}$ we get that

$$\tilde{\chi}_e(t) * \partial_t E(t)|_{t=(n+1/2)\Delta t} \approx \frac{E^{n+1} - E^n}{\Delta t}\tilde{\chi}_e^0 + \sum_{k=0}^{n-1} \frac{E^{n-k} - E^{n-k-1}}{\Delta t}\tilde{\chi}_e^{k+1}, \tag{2.222}$$

where

$$\tilde{\chi}_e^0 = \int_0^{\Delta t/2} \tilde{\chi}_e(\tau)d\tau \tag{2.223}$$

$$\tilde{\chi}_e^{k+1} = \int_{(k+1/2)\Delta t}^{(k+3/2)\Delta t} \tilde{\chi}_e(\tau) d\tau. \tag{2.224}$$

We will worry about actually evaluating these integrals shortly. For now, we note that we can take these results and use them to derive a time-stepping equation from Ampere's law. This will give us (in a semi-discrete form)

$$E^{n+1} = \left[E^n + \alpha (\nabla \times \mathbf{H})^{n+1/2} - \alpha \epsilon_0 \psi^n \right] \tag{2.225}$$

where

$$\alpha = \frac{\Delta t}{\epsilon_\infty + \epsilon_0 \tilde{\chi}_e^0} \tag{2.226}$$

$$\psi^n = \sum_{k=0}^{n-1} \frac{\tilde{\chi}_e^{k+1}}{\Delta t} \left(E^{n-k} - E^{n-k-1} \right). \tag{2.227}$$

Clearly, we can time step this equation within Yee's FDTD method quite easily so long as we can evaluate ψ^n efficiently.

Hence, we now need to consider more carefully how to efficiently evaluate the integrals in (2.223) and (2.224). As mentioned previously, the form of the electric susceptibility for the particular material being considered will influence how/whether a recursive convolution strategy can be used. For many practical media, we can utilize Debye or Drude-Lorentz-Sommerfeld models to determine the form of the electric susceptibility. For these kinds of materials, it is possible to expand their responses in a pole expansion of the form

$$\tilde{\chi}_e(t) = \sum_{p}^{N_p} a_p e^{-b_p t} u(t), \tag{2.228}$$

where $u(t)$ is the unit step function and depending on the material model being used a_p and/or b_p may be complex-valued numbers. Due to the properties of these simple materials, these poles always come in conjugate pairs so that another pole with values a_p^* and b_p^* will also be included in the summation in (2.228) so that the overall summation produces a real-valued function. Note that although it is possible to write the different material models into a form like (2.228), it may not be a trivial task to do this and determine the correct form of a_p and b_p based on material properties [14].

Now, assuming we have been able to arrive at an expression for $\tilde{\chi}_e(t)$ in a form like (2.228), we can evaluate the integrations in (2.223) and (2.224) to get

$$\tilde{\chi}_e^0 = \sum_{p=1}^{N_p} \frac{a_p}{b_p} \left(1 - e^{-b_p \Delta t/2} \right) \tag{2.229}$$

$$\tilde{\chi}_e^{k+1} = \sum_{p=1}^{N_p} \frac{a_p}{b_p} \left(e^{-b_p (k+1/2)\Delta t} - e^{-b_p (k+3/2)\Delta t} \right) = \sum_{p=1}^{N_p} \frac{a_p}{b_p} e^{-b_p (k+1/2)\Delta t} \left(1 - e^{-b_p \Delta t} \right). \tag{2.230}$$

Using (2.230) in (2.227), we get

$$\psi^n = \sum_{p=1}^{N_p} \left[\sum_{k=0}^{n-1} \frac{a_p}{b_p \Delta t} e^{-b_p(k+1/2)\Delta t} \left(1 - e^{-b_p \Delta t}\right) \left(E^{n-k} - E^{n-k-1}\right) \right]. \tag{2.231}$$

The sum within the square brackets can be evaluated recursively due to the simple exponential dependence on k. In particular, the previous values of the summation only need to be multiplied by $e^{-b_p \Delta t}$ to update them for the current time step. To see this, we can explicitly write out a few terms of ψ^n and group the terms carefully. For instance, we have for each pole in the expansion

$$\psi_p^1 = \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^1 - E^0\right) \tag{2.232}$$

$$\begin{aligned} \psi_p^2 = \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^2 - E^1\right) + e^{-b_p \Delta t} \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^1 - E^0\right) \\ = \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^2 - E^1\right) + e^{-b_p \Delta t} \psi_p^1 \end{aligned} \tag{2.233}$$

$$\begin{aligned} \psi_p^3 = \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^3 - E^2\right) \\ + e^{-b_p \Delta t} \left[\frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^2 - E^1\right) + e^{-b_p \Delta t} \psi^1 \right] \\ = \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^3 - E^2\right) + e^{-b_p \Delta t} \psi_p^2. \end{aligned} \tag{2.234}$$

This trend continues, allowing us to see that the general recursive evaluation takes the form of

$$\psi_p^n = \frac{a_p}{b_p \Delta t} e^{-b_p \Delta t/2} \left(1 - e^{-b_p \Delta t}\right) \left(E^n - E^{n-1}\right) + e^{-b_p \Delta t} \psi_p^{n-1}. \tag{2.235}$$

Evaluating (2.235) can be done very efficiently, and only requires a modest increase in computer storage (i.e., we need to store an additional past field value and the *recursive accumulator* ψ^{n-1}) and computation time.

Although this recursive convolution approach solves the basic problem of modeling dispersive media, there can still be complications for more complex situations. For instance, if a very broadband simulation of dispersive media is needed, many poles may need to be included in the Debye or Drude-Lorentz-Sommerfeld model potentially greatly increasing the total computation time. Additionally, for more complicated materials it may become more difficult to develop a suitable recursive computation process, if one can be found at all. Further, as mentioned previously, this method relied on a significant approximation that the field values and derivatives are constant functions over a time step. This is somewhat "standard" within the context of the FDTD method, but for more accurate results a more sophisticated representation of the fields may be needed. Some work has been done to improve this aspect of recursive convolution techniques within the FDTD method [13], but one may be pushed to using a more complicated numerical method in certain situations rather than trying to augment the FDTD method.

2.12.2 Auxiliary Differential Equation

Another popular approach to modeling dispersive materials is to formulate the FDTD equations using an additional differential equation (referred to as an *auxiliary differential equation*). This auxiliary differential equation is formulated to arrive at an intermediate time-stepping equation that allows us to avoid the direct computation of the convolution integral. This is relatively similar in principle to how we introduced auxiliary variables in treating the PML as an anisotropic absorber to avoid dealing with a complicated non-linear frequency dependence in a simple manner.

To see how this process works, we begin by noting that the second term on the right-hand side of

$$\partial_t D(t) = \epsilon_\infty \partial_t E(t) + \epsilon_0 \tilde{\chi}_e(t) * \partial_t E(t) \tag{2.236}$$

is the *polarization current* J_p of the material. Considering this, we can break this constitutive relation that will be substituted into Ampere's law into two equations as

$$\partial_t D(t) = \epsilon_\infty \partial_t E(t) + J_P(t) \tag{2.237}$$

$$J_P(t) = \epsilon_0 \tilde{\chi}_e(t) * \partial_t E(t). \tag{2.238}$$

We can rewrite (2.238) into the form of a differential equation that we can derive a time-stepping equation from depending on the form of $\tilde{\chi}_e(t)$. To perform this derivation, we convert (2.238) into the frequency domain as

$$J_P(\omega) = j\omega\epsilon_0 \tilde{\chi}_e(\omega) E(\omega). \tag{2.239}$$

At this point, we will need to consider a specific form for $\tilde{\chi}_e(\omega)$ to proceed.

To start, we will consider a simple material response. In particular, we will consider a single-pole Debye material of the form

$$\tilde{\chi}_e(\omega) = \frac{a_p}{j\omega + b_p}, \tag{2.240}$$

which has the additional property that a_p and b_p are real-valued numbers [15]. We can use this in (2.239) to get

$$(j\omega + b_p) J_P(\omega) = j\omega\epsilon_0 a_p E(\omega). \tag{2.241}$$

We can easily transform this back into the time domain to get

$$\partial_t J_P(t) + b_p J_P(t) = \epsilon_0 a_p \partial_t E(t). \tag{2.242}$$

This can be discretized at $t = (n + 1/2)\Delta t$ (the time step needed in Ampere's law) using central differences to get

$$\frac{J_P^{n+1} - J_P^n}{\Delta t} + b_p \frac{J_P^{n+1} + J_P^n}{2} = \epsilon_0 a_p \frac{E^{n+1} - E^n}{\Delta t}, \tag{2.243}$$

where we have used the average value to discretize the $J_p^{n+1/2}$ term that appears due to the second term of the left-hand side of (2.242). We can rearrange this for J_P^{n+1} to get

$$J_P^{n+1} = \frac{2\epsilon_0 a_p (E^{n+1} - E^n) + (2 - b_p \Delta t) J_P^n}{2 + b_p \Delta t}. \tag{2.244}$$

Now, when we go to actually discretize Ampere's law at $t = (n + 1/2)\Delta t$ we will need to evaluate

$$(\nabla \times \mathbf{H})^{n+1/2} = \epsilon_\infty (\partial_t E)^{n+1/2} + J_P^{n+1/2}. \tag{2.245}$$

Hence, we need to evaluate $J_P^{n+1/2}$ directly. We can utilize (2.244) to find that

$$J_P^{n+1/2} = \frac{J_P^{n+1} + J_P^n}{2} = \frac{\epsilon_0 a_p (E^{n+1} - E^n) + 2J_P^n}{2 + b_p \Delta t}. \tag{2.246}$$

We can substitute this result into Ampere's law and use a central difference for the time derivative to find that

$$E^{n+1} = E^n + \alpha (\nabla \times \mathbf{H})^{n+1/2} - \alpha \frac{2}{2 + b_p \Delta t} J_P^n \tag{2.247}$$

where

$$\alpha = \left[\frac{\epsilon_\infty}{\Delta t} + \frac{\epsilon_0 a_p}{2 + b_p \Delta t} \right]^{-1}. \tag{2.248}$$

Importantly, we can see from (2.244) that J_P^n can be computed from previous values of J_P and E so that (2.247) constitutes a valid time-stepping formula.

This method can be extended to handle other types of materials. However, the auxiliary differential equation used to compute the polarization current will often become more complicated. For instance, a Drude-Lorentz-Sommerfeld model material will lead to an auxiliary differential equation that involves second-order time derivatives of J_P and E. As a result, the formulation of a suitable discretization scheme and corresponding time-stepping equation becomes much more involved. Additionally, these higher-order derivatives also increase the memory consumption of the method since they require the storage of additional past data points of various quantities. Further, as with the recursive convolution method, handling materials with multiple poles in the expansion increases the computational cost of the method due to the need to include additional auxiliary differential equations. One area that the auxiliary differential equation method has some advantages in is that it can be extended to handle nonlinear materials quite readily [12, 15].

2.13 Far-Field Excitations and Results

Up to this point, we have typically assumed that any source of fields to our FDTD simulations was defined "locally" within the simulation region we are interested in. In particular, we always considered it to be represented by an electric current density (although local magnetic current densities could also be handled easily). For many situations, it is desirable to be able to consider the effect that a plane wave source has on our simulation; e.g., in calculating the radar cross section (RCS) of an object, analyzing certain antenna properties, etc. In principle, we could always place a current source very far away from our model so that the fields it produces look like a plane wave by the time they reach the simulation objects of actual interest. However, this would be extremely inefficient as it would require us to model the entire "empty space" in between the source and object of interest.

Instead of following this inefficient process, we will consider in this section how to directly excite a plane wave into our FDTD simulations. We will also briefly consider the related problem of computing far-field results from our FDTD simulations. In each case, we require some way to isolate the *scattered field* that is produced due to the presence of the inhomogeneous object we are studying. The particular mathematical description we use for these kinds of problems is to denote the total field $\mathbf{E}$ as

$$\mathbf{E} = \mathbf{E}_{\text{sc}} + \mathbf{E}_{\text{inc}}, \tag{2.249}$$

where $\mathbf{E}_{\text{inc}}$ is the incident field that would exist at any point in space if there were no inhomogeneity present and $\mathbf{E}_{\text{sc}}$ is the scattered field that is produced due to the inhomogeneity to ensure that $\mathbf{E}$ satisfies Maxwell's equations and the boundary conditions of the problem.

2.13.1 Plane Wave Excitation

We will begin by considering two approaches that provide a way to consider source excitations that are not explicitly included inside the simulation region and also allow for us to extract the scattered field from our results.

Scattered Field Method

The first approach is to formulate a FDTD method that directly uses the scattered field in its formulation. This can be done quite easily, and follows the principle of the volume equivalence principle. In particular, if we assume that we have some scattering object embedded in a homogeneous "background" region of free-space, then we can write Ampere's law by replacing $\mathbf{E}$ using (2.249) and similar for $\mathbf{H}$ to get

$$\nabla \times \big(\mathbf{H}_{\text{sc}} + \mathbf{H}_{\text{inc}}\big) = \epsilon \partial_t \big(\mathbf{E}_{\text{sc}} + \mathbf{E}_{\text{inc}}\big). \tag{2.250}$$

Since $\mathbf{E}_{\text{inc}}$ and $\mathbf{H}_{\text{inc}}$ satisfy Maxwell's equations in free-space (this is the definition of the incident field), we can easily determine that $\nabla \times \mathbf{H}_{\text{inc}} = \epsilon_0 \partial_t \mathbf{E}_{\text{inc}}$. We can use this in (2.250) to get

$$\nabla \times \mathbf{H}_{\text{sc}} = \epsilon \partial_t \mathbf{E}_{\text{sc}} + (\epsilon - \epsilon_0)\partial_t \mathbf{E}_{\text{inc}}. \tag{2.251}$$

We can treat the last term in (2.251) as a kind of equivalent current source since all terms involved are known for all time steps. Hence, we rewrite (2.251) as

$$\nabla \times \mathbf{H}_{\text{sc}} = \epsilon \partial_t \mathbf{E}_{\text{sc}} + \mathbf{J}_{\text{eq}}, \tag{2.252}$$

where $\mathbf{J}_{\text{eq}} = (\epsilon - \epsilon_0)\partial_t \mathbf{E}_{\text{inc}}$. We can follow a similar process for Faraday's law to get

$$\nabla \times \mathbf{E}_{\text{sc}} = -\mu \partial_t \mathbf{H}_{\text{sc}} + \mathbf{M}_{\text{eq}}, \tag{2.253}$$

where $\mathbf{M}_{\text{eq}} = (\mu - \mu_0)\partial_t \mathbf{H}_{\text{inc}}$. We can then discretize (2.252) and (2.253) using Yee's FDTD method to solve for the scattered fields produced by the incident field. This allows us to handle an arbitrary far-field excitation and inhomogeneous penetrable scatterers.

To deal with impenetrable scatterers like PEC objects, we must enforce our boundary condition in a slightly different way. In particular, we simply ensure the total field satisfies the correct boundary condition by modifying the typically homogeneous Dirichlet or Neumann boundary conditions to be inhomogeneous. For example, at a PEC object we would have

$$\hat{n} \times \mathbf{E} = 0 \rightarrow \hat{n} \times \mathbf{E}_{\text{sc}} = -\hat{n} \times \mathbf{E}_{\text{inc}} \tag{2.254}$$

or

$$\hat{n} \cdot \mathbf{H} = 0 \rightarrow \hat{n} \cdot \mathbf{H}_{\text{sc}} = -\hat{n} \cdot \mathbf{H}_{\text{inc}}. \tag{2.255}$$

Hence, we can fairly easily handle arbitrary problems in this way.

Although this method is straightforward, it has a significant drawback. In particular, we will have equivalent currents that must be utilized in our computation at every mesh cell within our inhomogeneous object. For large objects, this can lead to a non-negligible increase in total computation time. As a result, this method is not typically favored for practical FDTD implementations.

Total- and Scattered-Field Decomposition Method

This method is devised to address the shortcomings of the scattered field only method discussed in the previous section. It involves a hybrid approach between a standard FDTD method that solves for the total field and a FDTD method that solves for the scattered field only. These methods are applied in different regions of our problem, typically separated by a "fictitious" rectangular cube that we place around the simulation region we are interested in, but before we reach our ABC/PML region. By placing this fictitious boundary outside of the region where our actual object of interest is located, we can solve the scattered field only formulation of the FDTD outside of the cube and not need to consider any equivalent currents (a schematic of this is illustrated in Fig. 2.24). As a result, solving for the scattered field in this region doesn't really impact the computation time. Since inside our fictitious rectangular cube we solve the total field FDTD formulation, we also don't have to worry about any equivalent currents and also maintain a method without significant increase in computation time. Hence, this approach provides an elegant way to blend these two FDTD formulations together without costing a substantial increase in computation time.

The main complication with this approach occurs with augmenting our FDTD update equations near the fictitious boundary between the scattered and total field regions. However,

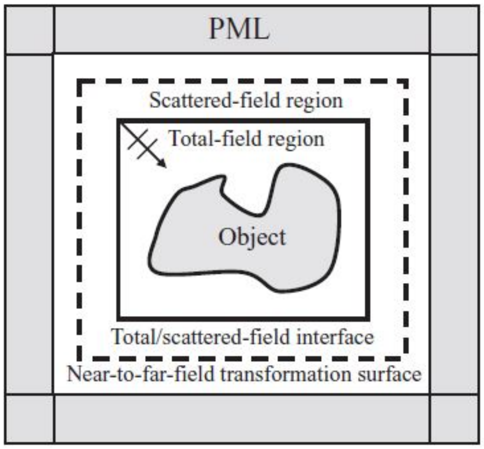

PML
Scattered-field region
Total-field region
Object
Total/scattered-field interface
Near-to-far-field transformation surface

Figure 2.24: Typical setup for the total- and scattered-field decomposition method (image from [5]).

this can be readily addressed and does not cause a significant change to the computational cost of the method. To see the basic process, we will consider a single surface at the interface between the total- and scattered-field regions. We will assume that this surface is perpendicular to the x-axis and is located at $i = I$. To the "left" of the surface ($i < I$) will be the scattered-field region and to the "right" of the surface ($i \geq I$) will be the total-field region. We will assume that the surface lies on the electric field grid, although the method can also be easily implemented for the magnetic field grid.

Now, for our update equations in the scattered-field region, we will come across situations where we need to use a data value that lies on the interface surface for which the data is stored as total fields. We then augment the scattered field time-stepping equations by simple replacements like

$$E^n_{\text{sc},y}(I, j+1/2, k) \to E^n_y(I, j+1/2, k) - E^n_{\text{inc},y}(I, j+1/2, k), \tag{2.256}$$

and similar for other electric field components. A similar process is also used in the total-field region update equations when a data value is needed but only the scattered field is stored at that location. In that situation, the incident field is added to the scattered field to recover the total field within the time-stepping equations.

Using this approach, the incident field is only needed to be known over a relatively small set of data points within the overall FDTD region. In principle, this approach would perfectly excite the desired far-field incident wave inside the total-field region and produce no incident field within the scattered-field region. However, because of the approximations of the discretization and the numerical dispersion, this is never truly the case. As a result, some special treatments have been developed to improve the performance of this approach to minimize unintentional leakage of the incident field into the scattered-field region.

2.13.2 Far-Field Results

When performing a simulation with a far-field source, we are often also interested in the results that are produced at far-field locations. We will now briefly discuss how these results can be computed using the FDTD method. The basic approach is to use the surface equivalence principle to define a set of equivalent currents that exist over a closed surface that completely encloses our simulation region of interest (this is usually referred to as a *Hugyens' surface*). Within the scattered- and total-field decomposition approach, this surface is placed in the scattered-field region but before the ABC/PML region (see Fig. 2.24). By computing the equivalent currents on this surface, we can then use a near-to-far-field transformation to compute the far-fields produced by the equivalent currents (which are defined in terms of the near-fields). In particular, these equivalent currents are computed as $\mathbf{J}_{\text{eq}} = \hat{n} \times \mathbf{H}_{\text{sc}}$ and $\mathbf{M}_{\text{eq}} = -\hat{n} \times \mathbf{E}_{\text{sc}}$.

Since these equivalent currents cover a closed surface, the uniqueness theorem ensures us that the fields produced by them outside of the surface can be made identical to the true exterior fields. To determine the fields, we use the surface equivalence principle to replace the interior region of our problem with a homogeneous background material. We can then follow a standard process to express the fields produced by the currents in terms of the electromagnetic potentials by convolving the free-space Green's function with the equivalent currents. This then becomes like an antenna analysis problem, where we can introduce the standard far-field assumptions to simplify these integrations. More details on how to implement this efficiently for FDTD analysis can be found in [2].

2.14 Source Temporal Profiles

Although we have discussed some basic details of how sources can be included in our FDTD simulations, we haven't commented on how these sources should vary as a function of time to perform a desired analysis. We will now briefly review some of the popular options for defining the temporal profile of sources.

One of the most basic temporal profiles is a Gaussian pulse. This is defined by

$$f(t) = \exp\left[-\frac{1}{2}(t/\tau_p)^2\right], \tag{2.257}$$

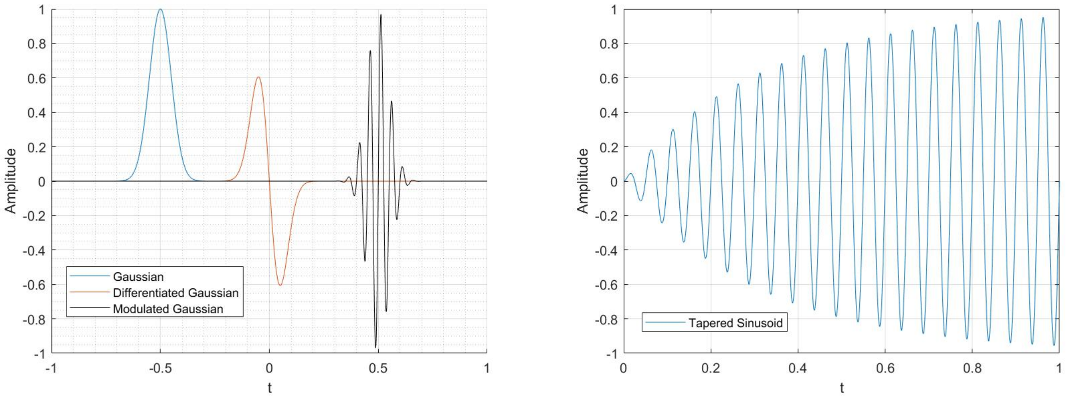
where τ_p sets the width of the pulse. Gaussian pulses are popular due to their simple mathematical form, convenient mathematical properties (e.g., the Fourier transform of a Gaussian function is another Gaussian function), and that they can represent a good approximation to some realistic pulse shapes that can be generated.

One drawback to using a Gaussian pulse is that it contains a DC component that can be problematic for some numerical simulations. To remove this DC component, a differentiated Gaussian pulse can be used that is defined by

$$f(t) = -\frac{t}{\tau_p}\exp\left[-\frac{1}{2}(t/\tau_p)^2\right]. \tag{2.258}$$

Another very popular alternative is to use a modulated Gaussian pulse, which is defined by

$$f(t) = \exp\left[-\frac{1}{2}(t/\tau_p)^2\right]\sin(\omega_0 t). \tag{2.259}$$



(a) (b)

Figure 2.25: Examples of common temporal profiles used for time domain simulations. Note that in (a) the various source distributions are shifted as needed so that they do not overlap with each other to make the visualization of their features easier.

This pulse retains the simple properties of a Gaussian pulse and provides a way to conveniently center the spectrum of the pulse at a desired center frequency ω_0.

Although we often think of using time domain methods to analyze a large bandwidth in a single simulation, there are situations where we desire to excite a time domain simulation with an approximately monochromatic source. An excellent example for this kind of situation would be in performing an analysis of nonlinear devices where we want to be able to easily analyze the production of higher-order harmonics of the input signal. In these situations, we can use a function like a tapered sinusoidal function defined by

$$f(t) = [1 - \exp(-t/\tau_p)] \sin(\omega_0 t). \tag{2.260}$$

The exponential taper is used to gradually increase the amplitude of the sinusoidal signal to reduce unintentional numerical noise/artifacts that can be produced by the discretization of the signal's temporal profile. Examples of some of these temporal profiles are plotted in Fig. 2.25.

It is also common to apply *windowing functions* to any of these temporal profiles to further control the shape of their spectral characteristics. Popular windowing functions include the Blackman-Harris and Taylor windows, however, almost any windowing function that is used in the signal processing field can also be used in numerical analysis. Often, the windowing functions are used to produce a sharper decline in the frequency content of the pulse outside of its specified bandwidth. However, this comes at the cost of increasing the "floor" of the frequency spectrum at higher frequencies. Whether this is valuable or not largely depends on a particular application area.

2.15 Finite Difference Method Project

This project covers the implementation of a computer code using the finite difference method to solve problems in electromagnetics. A list of suggested project topics are included later in this document. The main deliverable for this project will be a written formal report that details the work that was completed. At a high-level, this report will cover the formulation of the mathematical problem solved, the discretization approach used, and a discussion of the validation of the computer code via numerical results generated. A detailed grading rubric for this report is included later in this document.

2.15.1 Suggested Project Topics

1. Develop a 2D FDTD program using Yee's method to calculate the radiation of an infinitely long electric current in an open region that contains different inhomogeneities. The open region **must** be terminated using PMLs. After validating that the source radiates correctly in a homogeneous open region, use your code to study **at least two** of the following:

   (a) The diffraction pattern produced by an infinitely long current source radiating in the presence of an infinitely long conducting sheet with one slot. Compare the diffraction pattern of this case with that of an infinitely long conducting sheet with two or more slots.

   (b) The scattering produced when an infinitely long current source radiates in the presence of an infinitely long conducting cylinder of various cross sections (e.g., rectangular, circular, etc.).

   (c) The scattering produced when an infinitely long current source radiates in the presence of an infinitely long dielectric cylinder of various cross sections (e.g., rectangular, circular, etc.) and material properties.

   (d) Compare the performance of various approximate boundary conditions to terminate the open region. You should consider different PML parameters (thickness, conductivity profile, etc.) as well as two different ABCs (e.g., first- and second-order). **Completing this item can yield up to 5 points of extra credit to the total project score.**

2. Develop a 1D FDTD program to study the scattering of a plane wave from a dispersive or nonlinear material. You may use a PML or an ABC to terminate either or both of the ends of the simulation region. **Completing this item can yield up to 5 points of extra credit to the total project score.**

3. Solve Laplace's equation for various "shielded" transmission line structures that support TEM or quasi-TEM modes. Validate that your code is working by considering at least one geometry where reasonable analytical formulas exist for the line capacitance (e.g., a coaxial line or a stripline). For the geometries studied, plot the equipotential lines and static electric field distribution. For transmission lines that are not naturally "shielded" (e.g., a microstrip trace or a coplanar waveguide), ensure that the "extra" shield conductors are placed far enough away from the desired parts of the transmission line geometry that they minimally affect the solution. Possible transmission lines to study include: coaxial line, microstrip line, stripline, coplanar waveguide, grounded coplanar waveguide, slotline, etc.

4. Develop a 3D FDTD program using Yee's method to solve a wave propagation problem within a closed waveguide structure with regular shaped ports (e.g., rectangular cross sections that support well-known mode patterns). Terminate your ports using an absorbing boundary condition that can absorb the dominant mode of the waveguide at the location of the port (details on this boundary condition can be found about midway through Section 9.3.3 of your textbook, and does not require any knowledge of the finite element method to implement). **Completing this item can yield up to 10 points of extra credit to the total project score.**

5. Use the finite difference method to solve one problem of interest to you. Make sure to plan for some way to validate your code's performance for your selected problem.

2.15.2 Rubric

1. Title & Abstract (5 points)
   (a) Title and abstract are concise, but informative.
   (b) Abstract should properly convey the main information contained in the work, the methods used, and the problems studied.

2. Introduction and Conclusion (10 points)
   (a) Introduction should discuss relevant background and history of the problem to be studied and the methods used in the work, supported by relevant references from textbooks and the literature (around 4 or 5 references is likely plenty for this report). Introduction should also finish with a paragraph discussing the organization of the remainder of the paper.
   (b) Conclusion should succinctly summarize the content of the work and mention possible directions for further study, improvements that could be made to the numerical methods, etc.

3. Formulation & Discretization (30 points)
   (a) Equations that are to be solved numerically are appropriately derived from a well-established starting point (e.g., Maxwell's equations).
   (b) Assumptions or approximations of the derivation are clearly communicated.
   (c) Basic process of the numerical discretization is clearly communicated for all important/distinct equations. For example, you may need to show the derivation of a time-stepping formula in your report; if there is another time-stepping formula you use that is almost identical to the first one you don't need to show all the intermediary steps, just the final result.

4. Numerical Results (45 points)
   (a) Validation data is shown to demonstrate correct implementation of the numerical method. Sufficient details on the numerical results and validation data should

also be included so that someone else could conceivably implement their own tool and replicate your results. Sample items to cover would be sizes of the simulation region, spatial and temporal step sizes, kind of excitation waveform considered, relative permittivity and permeability of materials, etc. (Note: this is not an exhaustive list of what should be covered).

(b) Additional numerical results are presented to show utility of the numerical method. Again, sufficient detail is provided for simulation parameters that a reader can understand the content of the simulation and recreate it themselves.

(c) Figures are legible and aesthetically-pleasing (Matlab/Python plots are fine). Figure captions are concise, but informative. Figures are referenced and discussed appropriately within the text of the report.

(d) Note: your code must correctly implement the numerical method to approach reaching full points in this category of the rubric.

5. Writing Style (5 points)

   (a) Grammar, word use, spelling, etc. are of an overall good quality.

   (b) Best practices for writing mathematical prose are followed (equations are treated as part of the sentence, equations are numbered, "user-friendly" references to previous equations, etc.). See "What's Wrong with these Equations?" by N. David Mermin for basic guidelines to consider.

   (c) Equations are typeset in an aesthetically-pleasing manner.

   (d) Note: if the writing style is particularly poor, additional points will be subtracted from other aspects of the report (e.g., Formulation & Discretization or Numerical Results).

6. Coding Style (5 points)

   (a) Code is formatted and organized in an easily-readable manner. Descriptive variable and function names are used as appropriate.

   (b) Sufficient comments are used to make the code more easily interpreted by another person.

Chapter 3

Finite Element Method

3.1 Introduction to the Finite Element Method

We will now turn our attention to how the finite element method (FEM) can be used in computational electromagnetics analyzes. Just like the FDTD method, FEM is used to solve the PDEs that arise in various electromagnetic applications. As a result, there will be many high-level similarities between these two methods; e.g., the need to determine artificial boundary conditions to terminate open problems. Although these similarities exist, one of the primary advantages of FEM over FDTD methods is the improved capability for modeling complex geometries. The key to this capability is that FEM formulations approximate the *solution* to the PDE, while the FDTD method approximated the *differential operators*. It turns out that it is much simpler to develop more sophisticated approximations to the solutions of PDEs for relatively arbitrary geometries than it is to develop sophisticated approximations to differential operators. As a result, FEM formulations can utilize more realistic discretizations of complex geometries so that there are no staircasing geometrical errors. Correspondingly, if the FEM analysis is performed in a suitable manner the results can very frequently achieve excellent agreement with measured results for a fabricated device.

Although these improvements are very valuable, they do come with a cost. In particular, the theory and implementation of FEM formulations is typically more complex than corresponding finite difference methods. Further, FEM formulations generate matrix equations which cannot be solved in a trivial manner (e.g., the matrices are not purely diagonal). As a result, they must utilize various numerical linear algebra techniques to be solved. Due to the complexity of many electromagnetic problems, this step may not always be amenable to a trivial use of standard numerical linear algebra methods, and so must be carefully considered in the course of developing a robust FEM code. Further, the use of these more sophisticated numerical linear algebra techniques also typically increases the computational cost of the method compared to the FDTD method (although, direct comparisons to achieve a particular level of accuracy can be quite difficult, it is often problem specific which method would be more advantageous).

Another difference between FDTD and FEM is the amount of mathematical theory underpinning the two methods. The amount of theory underpinning the FDTD method is arguable, but for the most part we saw that it predominantly just involved Taylor series to build approximations to different derivatives. Other topics related to FDTD used various mathematical tools (e.g., the stability or numerical dispersion analysis), but these techniques were still relatively simple enough that we could introduce them almost immediately in class and work through them completely.

The story is quite different for FEM formulations, which have an extremely rich and detailed amount of sophisticated mathematical theory supporting the development of these methods. This detailed theory elegantly blends together topics from various mathematical fields to approach the fundamental questions of the existence and uniqueness of solutions to PDEs for realistic problems. This theory pulls on concepts from linear algebra, functional analysis, differential geometry, and more to provide the tools to rigorously analyze the properties of a PDE and determine the suitability of a proposed solution methodology. Although we will not go into detail on these advanced mathematical theories in this course, it is important to note that the finite element method fits extremely naturally into this analysis framework. As a result, it is not uncommon to see certain concepts from these mathematical analysis techniques appear occasionally in engineering papers on CEM. We will introduce some simple concepts and terminology as appropriate from this more mathematical approach to discussing FEM formulations, but we will not leverage them to any significant depth in this course.

3.2 Basic FEM Process

We will now look at the basic FEM process for taking a continuous PDE and converting it into a linear matrix equation. There are a variety of ways to go about formulating the matrix equation, but we will only look at a fairly general but simple approach currently. This approach is sometimes referred to as the *weighted residual method*. It shares many similarities with the concept of formulating a *weak form* of a PDE or the corresponding *weak solution* to the PDE. We will expand on the origin of these terms shortly when they will make more sense due to the context.

When dealing with complicated PDEs (or integral equations, which we will learn about later in this course), it is common to adopt an *operator notation* as a shorthand to simplify writing equations. In this form, a typical PDE will be written symbolically as

$$\mathcal{L}\varphi = f, \tag{3.1}$$

where $\mathcal{L}$ is the differential operators defining the PDE, φ is the solution to the PDE, and f is the driving function that acts as a source to the PDE. This $\mathcal{L}$ can take on various forms, e.g.,

$$\mathcal{L} = \nabla^2 \tag{3.2}$$

for Poisson's equation or

$$\mathcal{L} = \nabla \times \nabla \times + \mu\epsilon\partial_t^2 + \mu\sigma\partial_t \tag{3.3}$$

for the wave equation in a lossy medium with constant conductivity. Clearly, an expression like (3.2) or (3.3) only makes sense when we think of it being applied to a function with suitable properties such that the various derivatives can be evaluated in a meaningful manner.

Now, to begin to convert (3.1) into a finite-dimensional matrix equation we will first need to come up with a way to approximate φ using a discrete set of variables. We do this by expanding the unknown function φ with a set of known *basis functions* (also sometimes referred to as *expansion functions*) that have unknown expansion coefficients. As an example, consider a one-dimensional problem along the x-axis. Then, we expand $\varphi(x)$ as

$$\varphi(x) \approx \sum_{j=1}^{N} c_j v_j(x), \tag{3.4}$$

where c_j is the unknown expansion coefficient and $v_j(x)$ is a known continuous basis function. The exact form that $v_j(x)$ should take is problem-specific. However, a general rule will be that $v_j(x)$ should be able to satisfy the boundary conditions of the problem being considered and that it can make a good approximation to $\varphi(x)$ throughout the spatial domain of interest. Overall, the choice of $v_j(x)$ has an incredibly important impact on the performance of a FEM formulation and must be selected carefully. We will consider various options for different types of problems as we go through this section of the course.

With (3.4) in hand, we now have a discrete number of variables that we need to solve for (i.e., all the c_j's). If we substitute (3.4) into (3.1), we are faced with the problem that we still have an infinite-dimensional problem due to the infinite x values that we need to have our PDE enforced at. It should not be too surprising at this stage that it is somewhat hopeless to try and use (3.4) with a discrete number of expansion functions to perfectly solve (3.1) at all x. This kind of solution is often called a *strong solution*, since it would perfectly satisfy the PDE at all points of space. In general, finding a strong solution to a PDE is almost impossible except for a few very specific problems such as solving for the scattering from a sphere or other problems that you considered exactly in an electromagnetic theory course.

Instead, we will now search for the *weak solution* to the PDE. What this means is that we are going to relax our requirements for what we will consider to be a *solution* to the PDE. The typical way to do this is to enforce the PDE in some kind of *averaged sense* by multiplying (3.1) by what is known as a *testing function* or *weighting function* and then integrating this over the spatial domain of interest. For a particular weighting function w_i, we can put all of these steps together to get

$$\int w_i(x)\mathcal{L}\left(\sum_{j=1}^{N} c_j v_j(x)\right)dx = \int w_i(x) f(x) dx. \tag{3.5}$$

We can repeat this process for a sufficiently large set of w_i's to get a system of linear algebraic equations that can be solved to find the c_j's. The approach illustrated in (3.5) is also the origin of the name *weighted residual method*. We can move all terms to one side of this equation and see that our process is equivalent to minimizing the residual error of our approximation, i.e., $\mathcal{L}[\sum_{j=1}^{N} c_j v_j(x)] - f(x)$, in a *weighted* sense due to our testing/weighting function w_i.

Again, it should hopefully not be too surprising that we must be very careful with how we go about choosing our set of w_i's to ensure that our resulting matrix equation has "good" properties so that it can be solved numerically. Much of the advanced mathematical theory alluded to earlier is focused on analyzing the properties of $\mathcal{L}$ for particular combinations

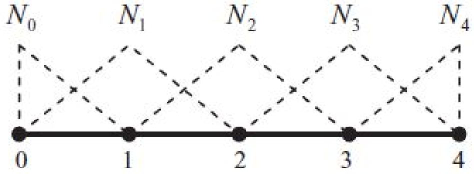


Figure 3.1: Example of linear interpolating basis functions for a 1D problem (image from [5]).

of v_j's and w_i's to determine whether the resulting weak solution can be found, if it will be well-behaved, and other properties of the equation being studied. Understanding these properties often gives significant insight into the properties that the resulting matrix equation will have, e.g., how difficult it may be to solve it. Hence, this deeper mathematical theory is often extremely valuable in developing improved numerical methods for solving PDEs.

We will now briefly consider the question of how to determine the v_j's and w_i's to solve a particular PDE. Up to this point, we have not restricted their properties in any way beyond somewhat vague statements about the functions needing to be sufficiently differentiable so that we can evaluate the expressions in (3.5) and that the functions should provide a "good" approximation to the known behavior of the function being expanded (e.g., be able to satisfy some kind of boundary condition). Typically, finding a function that satisfies these properties on a global scale (i.e., over all values of x that are of interest in the problem) for a general/arbitrary 3D electromagnetic problem is extremely challenging. However, this task is significantly simpler if we make each $v_j(x)$ only need to expand $\varphi(x)$ over a fairly small spatial range. This is the core idea of the finite element method; i.e., to expand $\varphi(x)$ with a set of simple functions that each only have a small spatial support, but as an entire set are able to faithfully represent $\varphi(x)$ over all x of interest to some desired level of accuracy. An example v_j for a simple 1D problem is shown in Fig. 3.1, which provides a linear interpolation accuracy. Obviously, using these functions requires us to break our overall problem space up into enough "finite elements" that having a linear approximation to $\varphi(x)$ over each element provides a reasonable accuracy.

Once a v_j has been selected, there are many different ways to go about choosing a suitable w_i. This is where the more advanced theory about PDEs becomes useful. In particular, we can think of the solutions to our PDE as forming a type of vector space, typically referred to as a *function space*. Our PDE can then be analyzed (using what is often referred to as functional analysis) as a kind of "transformation" or "map" between different function spaces in similarity to how we would analyze different kinds of linear operators/maps in linear algebra. We can then define the domain and range spaces of our PDE. To develop a good weak solution to our PDE, it is essential that we choose our basis functions so that they exist in the domain space of our PDE. Likewise, we should choose our testing functions so that they are *related* to the range space of the PDE. Often, we should not choose our testing functions directly from the range space, but rather from what is known as the *dual space* to the range space (we will discuss this more later). For many electromagnetic PDEs, it works out that the dual space to the range space of the PDE actually matches the domain space of the PDE (there are some very deep mathematical reasons for this related to the concepts of an operator being self-adjoint or Hermitian, which happens quite frequently in physics, but not always). Hence, we can often get a "good" discretization of our PDE by choosing our w_i's to match the v_j's. This approach is often referred to as *Galerkin's method*. It is not uncommon to come across papers in the literature erroneously claiming that Galerkin's method is always a good option/idea, but this is a naive and incorrect statement that is slowly fading out of popular thought.

When we use Galerkin's method, we are finally able to convert our PDE in (3.1) into a matrix equation that can be solved numerically. We have

$$[\mathbf{L}]\{\mathbf{c}\} = \{\mathbf{f}\}, \tag{3.6}$$

where

$$[\mathbf{L}]_{ij} = \int v_i(x)\mathcal{L}v_j(x)dx, \tag{3.7}$$

$$\{\mathbf{c}\}_j = c_j, \tag{3.8}$$

$$\{\mathbf{f}\}_i = \int v_i(x)f(x)dx. \tag{3.9}$$

We often must use numerical integration routines (typically referred to as quadrature methods/rules) to evaluate the different integrals in (3.7) and (3.9), although analytical evaluations are possible for certain restricted scenarios (that are still of practical interest).

3.3 FEM Analysis: 1D Case

Previously, we covered the general idea of how we can go about using the finite element method to formulate a solution to a PDE. We will now take a closer look at some of the finer details that one must address when actually solving a particular problem. To keep the presentation simple, we will begin by doing this for a 1D case where we can use very simple basis functions and notation. We will eventually extend our process to higher dimensions and to more complicated electromagnetics problems to see how FEM can be used to solve more realistic problems.

As with previous cases, finding a suitable 1D electromagnetics problem can be somewhat challenging. Relevant examples would include a normal incidence plane wave propagating through a planar-layered medium or a transmission line problem. Since our goal is to only illustrate the basic FEM process, we will consider the simplest case of a normal incidence plane wave propagating in a homogeneous medium with different boundary conditions presenting the inhomogeneity to the problem. For this situation, we will have the Helmholtz equation of

$$\frac{d^2}{dx^2}E_z(x) + k^2E_z(x) = f(x), \quad 0 < x < L. \tag{3.10}$$

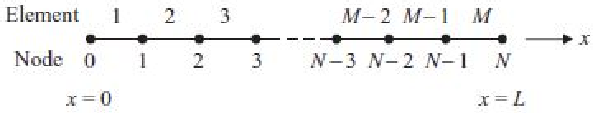


Figure 3.2: Subdivision of the solution domain into finite elements and the nodes in between elements (image from [5]).

Our PDE is not fully specified without also providing the boundary conditions. We will assume that we have a Dirichlet boundary condition of

$$E_z|_{x=0} = p \tag{3.11}$$

and a Robin boundary condition of

$$\left[\frac{d}{dx}E_z + \gamma E_z\right]_{x=L} = q. \tag{3.12}$$

This Robin boundary condition can be used as a Neumann condition (if $\gamma = 0$), or could be used to model a kind of impedance boundary condition or ABC for our particular problem. The specifics for this artificial problem are unimportant, the main point is how we will handle incorporating this boundary condition into developing our FEM matrix equation. Overall, (3.10) to (3.12) constitute a complete specification of the PDE, which we can now go about solving with the finite element method.

To begin, we will need to discretize our solution domain of $0 \leq x \leq L$ into a set of smaller subdomains that we can define simple basis functions over. For a 1D problem, these smaller subdomains will just be short line segments that will serve as the finite elements for our solution. At the intersection between two elements, we will have a *node*. For a simple 1D problem, we can number the elements and nodes in a fairly straightforward manner (see Fig. 3.2). However, for higher dimensions, there won't exist an obvious or unique numbering scheme so we will have to be more careful with determining our conventions and how we store this information in our code (this is usually referred to as a *connectivity list*, which we will discuss more later in the course). One important detail to note about our subdivision is that it can be *non-uniform*; i.e., each element can have a different length without causing any issues/difficulty within the FEM formulation. As a result, we can optimize how long the elements are within different regions of the problem to use a minimal number of elements to accurately represent the solution (this often must be done adaptively for practical problems, and is known as adaptive mesh refinement). This is an important distinction compared to finite difference methods that we considered previously, which typically worked on uniform grids or could only change the non-uniformity slowly along a single dimension at a time.

Before we choose which basis and testing functions to use in our FEM formulation, it is useful to perform some rearranging of our PDE into what is more commonly referred to as its *weak form*. To do this, we will test the PDE in (3.10) with an (at this point) unspecified testing function $w(x)$ so that we have

$$\int_0^L w(x)\left[\frac{d^2}{dx^2}E_z(x) + k^2 E_z(x)\right]dx = \int_0^L w(x)f(x)dx. \tag{3.13}$$

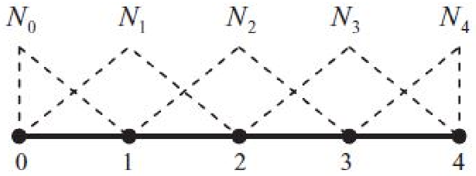


Figure 3.3: Linear interpolating functions for use in a 1D FEM analysis (image from [5]).

One of the main benefits of having the testing function is, if it is smooth enough, we can transfer some of the derivatives that are currently applied to E_z onto the testing function by using integration by parts. This will lower the smoothness needed for any basis function used to expand E_z, and is generally a very useful advantage in formulating numerical solutions. Doing this integration by parts, we get

$$\int_0^L \left[\frac{dw(x)}{dx}\frac{dE_z(x)}{dx} - k^2 w(x)E_z(x)\right]dx - \left[w(x)\frac{dE_z(x)}{dx}\right]_{x=0}^{x=L} = -\int_0^L w(x)f(x)dx. \quad (3.14)$$

This is the weak form of the PDE given in (3.10). Note that the boundary conditions given in (3.11) and (3.12) are still needed in our specification of the problem, and we will use them shortly as we proceed with the FEM discretization of this problem.

We can now choose what functions to use as basis and testing functions. Due to the symmetry of (3.14), the Galerkin method is a popular discretization approach. Further, because we need to evaluate the spatial derivative of our basis and testing functions, we will want to at least use some kind of linear *interpolating function*. Typically, an interpolating function is defined so that it takes on a value of 1 at a particular "data point" and varies (typically along some polynomial order) to a value of zero at all other "data points" (for this problem a "data point" will be a node of the mesh, but other FEM analyses will have different kinds of "data points"). This helps simplify the formulation and solution of the interpolation problem for a particular data set. A simple example of linear interpolating functions for our 1D problem is shown in Fig. 3.3, which corresponds to a set of *triangular functions*. The mathematical specification of these functions is simple for a uniform mesh, but is more involved for a non-uniform mesh. We will consider the more complicated case of defining these interpolating functions for an arbitrary mesh later.

As suggested by Fig. 3.3, we will use full triangular functions at all of the interior points of our mesh. However, to handle the boundary conditions, we will need to use "half-basis functions" at the two extreme edges of the mesh. If we label each basis function by the node number it is attached to (e.g., $N_4(x)$ is the triangle function centered at node 4 of the mesh), then we can expand our unknown function E_z as

$$E_z(x) = \sum_{j=0}^{N} a_j N_j(x). \quad (3.15)$$

However, we can actually simplify this expansion right away due to the properties of the interpolating functions and the presence of our Dirichlet boundary condition. In particular, we see from (3.11) that the expansion coefficient for N_0 will have to be equal to the Dirichlet boundary condition data. That is, we have $a_0 = p$ so that our full set of unknown expansion coefficients will be slightly reduced due to the already known information from the boundary condition. Hence, we will actually have that

$$E_z(x) = \sum_{j=1}^{N} a_j N_j(x) + pN_0(x). \tag{3.16}$$

Before substituting this expansion into (3.14), we need to do a little more work with the boundary term that arose from our integration by parts (this is the second set of square brackets in the equation). Our first change will be to note that the set of testing functions we will be using are all N_i from the set of $i = 1, 2, \dots N$. Note that $i = 0$ is specifically excluded from this set because this is not a basis function that we need to solve for (due to the Dirichlet boundary condition), and so including it in the testing set would lead to a non-square matrix equation that we cannot solve uniquely (this is typically avoided in the CEM field since physics problems should have unique solutions). Due to this, the portion of the integration by parts at $x = 0$ is no longer needed in our tested equation because all $N_i(0) = 0$ due to the properties of this set of interpolating functions. *(Note: the effect of this boundary condition will still be felt in the overall FEM matrix equation, as we will see shortly.)* Hence, we have that

$$\int_0^L \left[\frac{dN_i(x)}{dx}\frac{dE_z(x)}{dx} - k^2 N_i(x)E_z(x)\right]dx - \left[N_i(x)\frac{dE_z(x)}{dx}\right]_{x=L} = -\int_0^L N_i(x)f(x)dx. \tag{3.17}$$

We can now use our Robin boundary condition data given in (3.12) to rewrite this as

$$\int_0^L \left[\frac{dN_i(x)}{dx}\frac{dE_z(x)}{dx} - k^2 N_i(x)E_z(x)\right]dx - \Big[N_i(x)(q - \gamma E_z)\Big]_{x=L} = -\int_0^L N_i(x)f(x)dx. \tag{3.18}$$

From here, we can substitute the expansion from (3.16) into (3.18) to get

$$\sum_{j=1}^{N} a_j \int_0^L \left[\frac{dN_i(x)}{dx}\frac{dN_j(x)}{dx} - k^2 N_i(x)N_j(x)\right]dx - \left[N_i(x)\big(q - \gamma\sum_{j=1}^{N} a_j N_j(x)\big)\right]_{x=L}$$
$$= -\int_0^L N_i(x)f(x)dx - p\int_0^L \left[\frac{dN_i(x)}{dx}\frac{dN_0(x)}{dx} - k^2 N_i(x)N_0(x)\right]dx. \tag{3.19}$$

(Note: the final term on the right-hand side of (3.19) comes from substituting our known quantity from the Dirichlet boundary condition into the weak form and rearranging. This is how the effect of the Dirichlet boundary condition still manifests itself into the overall solution of the problem.)

This can be assembled into a matrix equation for the a_j's as

$$[\mathbf{K}]\{\mathbf{a}\} = \{\mathbf{b}\} \tag{3.20}$$

where

$$[\mathbf{K}]_{ij} = \int_0^L \left[\frac{dN_i(x)}{dx}\frac{dN_j(x)}{dx} - k^2 N_i(x)N_j(x)\right]dx + \gamma\delta_{iN}\delta_{jN}, \tag{3.21}$$

and δ_{ij} is a Kronecker delta function. The expression for $\{\mathbf{b}\}$ is somewhat more complicated due to the various pieces of (3.19) that can contribute to it. In particular, we have the data dependent on $f(x)$, as well as the q- and p-dependent terms. Putting all of these together, we get

$$\{\mathbf{b}\}_i = q\delta_{iN} - p\int_0^L \left[\frac{dN_i(x)}{dx}\frac{dN_0(x)}{dx} - k^2 N_i(x)N_0(x)\right]dx - \int_0^L N_i(x)f(x)dx. \tag{3.22}$$

This process constitutes the more complete steps involved in using the finite element method to discretize a differential equation. Before moving on, there are a few important points to be made about the matrix equation in (3.20). First, due to the limited support of the basis and testing functions, the matrix $[\mathbf{K}]$ is extremely sparse (i.e., most elements are zero). This is important because many practical problems have matrix sizes that can become extremely large (e.g., millions to billions in dimension). Due to the prevalence of sparse matrices in practical applications, efficient data structures have been developed to only store the non-zero elements of a sparse matrix to greatly lower the amount of computer memory needed to represent the matrix. Computational routines to efficiently work with sparse matrices have also been developed, making solving large FEM problems with relatively modest computational resources quite feasible. Another "nice" feature of $[\mathbf{K}]$ is that it is symmetric. Symmetric matrices have been studied extensively in linear algebra, and so there is a wealth of information available about the properties of symmetric matrices and even in some cases numerical routines can be optimized to work specifically on matrices of this kind.

Finally, we will comment briefly on how to actually evaluate the integrals in (3.21) and (3.22). Due to the simplicity of the functions used, it is possible to evaluate most of these integrals analytically if we assume that material properties are constant over each finite element. Formulas for this can be found in [5, Sec. 9.1.2]. Assuming this for the driving function $f(x)$ may not always be reasonable. In this situation, we can utilize a number of different numerical quadrature techniques to evaluate the integral numerically. A very popular general purpose numerical quadrature technique is known as *Gaussian quadrature* or *Gauss-Legendre quadrature*. These quadrature rules approximate the definite integral of a function as a weighted sum of function values at specified points within the domain of the integration. For example, we have

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{N} w_i f(x_i). \tag{3.23}$$

The x_i are the sample points and the w_i are the weighting values. The specific values for x_i, w_i, and N depend on the particular quadrature rule being used and the order of the quadrature rule. Gaussian quadrature is popular due to its simplicity and because it is designed to exactly integrate polynomials of degree $2N-1$ for an N-point rule. Over the small dimensions of finite elements, a polynomial approximation to a function like $f(x)$ is typically reasonable, and so these quadrature rules can be very accurate and relatively efficient.

3.4 A (Very) Brief Introduction to Function Spaces

When we arrived at our weak form of the Helmholtz equation given in (3.14) we somewhat quickly jumped to suggesting a particular set of basis and testing functions to continue with the FEM process for solving the differential equation. We will now take a little bit more time to discuss some mathematical topics behind selecting these functions. In particular, we will focus on introducing a few function space concepts that come up somewhat frequently in the CEM literature.

Recall that our weak-form equation was

$$\int_0^L \left[\frac{dw(x)}{dx}\frac{dE_z(x)}{dx} - k^2 w(x)E_z(x)\right]dx - \left[w(x)\frac{dE_z(x)}{dx}\right]_{x=0}^{x=L} = -\int_0^L w(x)f(x)dx. \quad (3.24)$$

We can readily see that we need to choose a basis function for E_z and a set of testing functions for w that will allow us to meaningfully integrate all the terms in (3.24). Due to the symmetry of the overall equation, we also see why the Galerkin method is so popular for many PDEs.

Since derivatives make functions less smooth (and therefore more prone to producing non-integrable features), we can typically think of the first term in (3.24) as presenting the most difficult term to integrate. Hence, we will need to ensure that we choose basis and testing functions so that their derivatives are square integrable, i.e., that

$$\int_0^L \left|\frac{dN_i(x)}{dx}\right|^2 dx < \infty, \quad (3.25)$$

where N_i would be a basis/testing function. In the more mathematical theory of PDEs and FEM, we would typically say that we need our functions to be members of the L^p function space with $p = 2$ (the L stands for Lebesgue, so these are also sometimes referred to as Lebesgue spaces). For a function to be in an L^p space, the following norm must be finite

$$||g||_p = \left(\int |g|^p dx\right)^{1/p} < \infty. \quad (3.26)$$

These L^p spaces are a special example of *Banach spaces* (i.e., a complete normed vector space) that are particularly useful in signal processing and optimization applications. Banach spaces are particularly useful because they have a norm that can serve as a *metric* in these abstract function spaces. The metric gives us a way to define the length of a vector/function, and can also allow us to compute the "distance" between functions (i.e., how different they are from each other). These operations are vital in the formulation and solution of many practical engineering problems, and is one of the reasons why the theory of these spaces is so valuable in engineering.

For physical systems, we are typically most interested in what are known as L^2 functions since the energy in many physical systems can be related to integrals like (3.25). For instance, we have in electromagnetics that

$$\text{Energy } = \int \frac{1}{2}\Big(\epsilon|\mathbf{E}|^2 + \mu|\mathbf{H}|^2\Big)dx, \quad (3.27)$$

so that both **E** and **H** need to be square-integrable to define a set of fields with finite energy. Due to this, people with more of a physics background will often refer to functions in L^2 as being "finite energy" functions (or some variant of those words).

One special property of the L^2 space is that it is not only a Banach space, but is also a *Hilbert space* (i.e., a complete inner product space). Having access to an inner product gives us all of the benefits of a Banach space, but also provides us with the added ability to define "angles" between abstract functions/vectors. This is *extremely valuable* in analyzing complicated and abstract situations, since it allows us to extend much of our geometric intuition about problems in three dimensions to infinite-dimensional spaces. For instance, in a Hilbert space we can find an orthonormal basis for the space that can be very useful in solving problems (e.g., Fourier theory fits within this Hilbert space viewpoint). Along these lines, the inner product for the L^2 space is given by

$$\langle g, h\rangle = \int h^*(x)g(x)dx, \tag{3.28}$$

where the $*$ denotes a complex conjugate. Since our testing functions will often be real-valued in FEM analysis, we can think of an equation like (3.13) as being the inner product between the PDE and our testing function. That is, we are seeing how much of the PDE "aligns" with a particular testing function. Alternatively, we are projecting our solution onto a particular vector/function within our Hilbert space.

Considering this, we will sometimes write out an equation like (3.13) in a convenient shorthand as

$$\langle \mathcal{L}E_z, w\rangle = \langle f, w\rangle, \tag{3.29}$$

where $\mathcal{L}$ is the differential operator defining the PDE. Often, the CEM community will reverse the order of these arguments to write (3.13) as

$$\langle w, \mathcal{L}E_z\rangle = \langle w, f\rangle, \tag{3.30}$$

where now the complex conjugate would be applied to the first argument of the inner product rather than the second as is done in the mathematical convention of (3.28).

Now, the final kind of function space we need to introduce is a *Sobolev space*. In general, a Sobolev space is a kind of Banach space that requires a norm that involves a function and some number of derivatives of the function to be finite. The theory of Sobolev spaces is much more general than we need to go into for our purposes, so we will only focus on a few special cases throughout this course that are of interest to CEM (and other areas of physics). In many physics problems, our Sobolev spaces will not just be Banach spaces, but will also be Hilbert spaces (i.e., they have an inner product in addition to a norm). As an example, the Sobolev space H^1 would be defined as functions that satisfy the following inequality:

$$\begin{aligned} ||f||_{H^1} &= \left(||f||_2^2 + \left|\left|\frac{d}{dx}f\right|\right|_2^2\right)^{1/2} \\ &= \left[\int \left(|f|^2 + \left|\frac{d}{dx}f\right|^2\right)dx\right]^{1/2} < \infty \end{aligned} \tag{3.31}$$

Comparing this to the integrations required to be evaluated in (3.24), we see that this is exactly the space that we need to be choosing our basis functions from. As a result, the more mathematical theory of PDEs is often specified in terms of Sobolev spaces. In particular, it specifies which Sobolev spaces serve as the domain and range spaces of a particular differential operator. We can use this information to help formulate good numerical discretization strategies. It has been found repeatedly in the CEM literature that numerical discretization strategies that *conform* to the Sobolev space properties of the underlying weak-form differential (or integral) equation result in better performing numerical methods than other choices of basis and testing functions [16–18].

3.5 Scalar Basis Functions in Higher Dimensions

Having looked at a particularly simple 1D problem, we are now interested in developing FEM formulations for more realistic problems in higher dimensions. To keep the process simple, we will initially focus on scalar-valued problems in electromagnetics. For instance, examples of this occur in 2D, as well as for 3D electrostatic problems (e.g., Laplace's or Poisson's equation). Before considering the actual formulation of these problems, we will discuss the development of linear interpolating functions that can serve as basis and testing functions for these problems.

The first step in developing the linear interpolating function is to specify the shape of the finite elements we will be using. In principle there are many different possible choices, but in the CEM community it has been found that triangular (in 2D) or tetrahedral (in 3D) elements typically lead to the best numerical methods (examples of these meshes are shown in Fig. 3.4). This is due to a combination of these elements being quite flexible at modeling complex shapes and because the CEM community has been able to develop basis functions with good properties (i.e., they can represent EM quantities well) for these elements. Although this is the typical approach in the CEM community, it is not uncommon to find other kinds of elements in wide use in other areas of physics/engineering. These other approaches can be adapted for use in CEM, although this is not particularly common.

We will now go about finding a suitable mathematical definition for a linear interpolating function on a triangular mesh. We will follow a different process than that shown in [5] because it is a much more compact derivation. However, the method shown in [5] can be more straightforward for extending to the consideration of higher-order interpolating functions.

To begin, we will recall what properties we desire for our linear interpolating function. First, the function should be equal to 1 at a particular "data point" it is associated with and then linearly varies to 0 at all adjacent "data points". For the triangular (and tetrahedral) meshes, the "data points" for scalar basis functions will typically be the *nodes* of the mesh. As a result, our linear interpolating functions will look like a higher-dimensional version of the triangle function used in our 1D analysis. This kind of function is sometimes referred to as a pyramidal function, and is illustrated in Fig. 3.5.

From Fig. 3.5, it is clear that the easiest way to define this function will be as a piecewise combination of functions defined over each individual triangle of the mesh. Towards this purpose, we will focus on a single triangle and look for 3 different functions that are equal

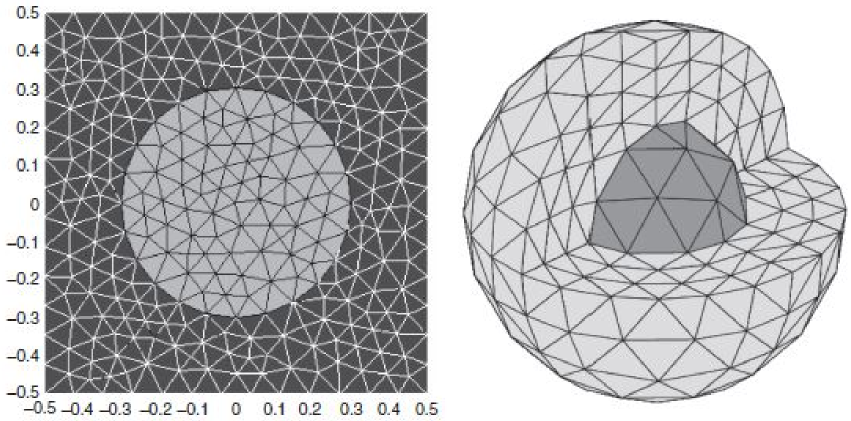


Figure 3.4: Finite element meshes for triangular (left) and tetrahedral (right) elements. Only the external surface of the tetrahedral elements are shown for clarity (images from [5]). Note how these meshes can do a good job representing curved surfaces with relatively low error compared to the staircasing errors that occur in finite difference analysis.

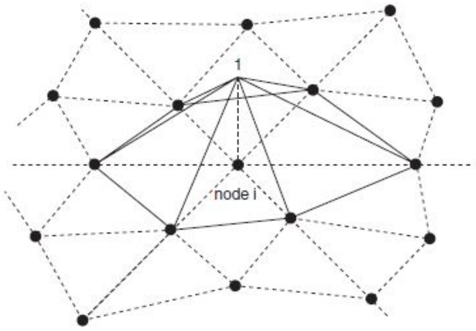


Figure 3.5: Illustration of a linear interpolating function on a triangular mesh that is associated with a particular node of the mesh (image from [5]).

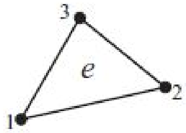


Figure 3.6: Triangular element with a set of local node numbers (image from [5]).

to 1 at a single node of the triangle and linearly decay to 0 at all the other nodes of the triangle. To help with bookkeeping, we will always establish a set of *local node numbers* for each node of a triangular element, as shown in Fig. 3.6.

We can determine the needed linear interpolating functions by setting up and solving a simple system of equations. However, this is rather tedious. The simplest way to derive the linear interpolating functions are to look at the desired properties of them, and then determine which mathematical objects already possess many of these properties. As mentioned previously, the defining properties of this function is that it be linear, be equal to 1 at the node it is defined at, and is zero at all other nodes. This creates a basis function with a finite support that causes it to only be non-zero over elements that the node is connected to, which leads to the sparsity of the FEM system matrix.

From these properties, it is seen that a properly structured determinant can accomplish these goals. A more sophisticated view of the definition of a determinant shows that it is a skew-symmetric multilinear function of the columns (or rows) of the matrix. This linearity helps ensure that when we use the determinant to define our interpolating functions, they will be linear functions, since as will be seen shortly only one column is not constant. Further useful properties of the determinant are that if any two columns or rows are identical, then the result will be zero. This allows us to construct the basis function as being the determinant of a 3x3 matrix whose entries include the locations of the nodes for a given element. The basis function for a specific node is then formed by replacing the column or row that holds the coordinates of that node and replacing them with the variables x and y. As an example, a possible general form for the basis function at the first local node of an element e is

$$N_1^{(e)} = \begin{vmatrix} 1 & 1 & 1 \\ x & x_2 & x_3 \\ y & y_2 & y_3 \end{vmatrix}, \tag{3.32}$$

which has the desired properties of linearity and equaling 0 if $(x, y) = (x_2, y_2)$ or (x_3, y_3), the locations of nodes 2 and 3, respectively. However, there is no guarantee that the basis function will equal 1 at $(x, y) = (x_1, y_1)$, and so the following normalization is used to enforce this property, resulting in the basis function at node 1 being

$$N_1^{(e)} = \begin{vmatrix} 1 & 1 & 1 \\ x & x_2 & x_3 \\ y & y_2 & y_3 \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}. \tag{3.33}$$

This basis function, although defined differently, exactly matches the one shown in [5], and generalizes to tetrahedral elements very easily.

We can use these linear interpolating functions to define the overall pyramidal function by using proper bookkeeping of the relationship between the global node number that the pyramidal function is associated with and the local node number it corresponds to in each of the triangular elements the node is attached to. We will denote the overall pyramidal basis function as N_j, where j is the global node number of the basis function. This basis function will be composed of a summation of one of $N_1^{(e)}$, $N_2^{(e)}$, and $N_3^{(e)}$ for each triangular element e that the global node is attached to.

3.6 Scalar FEM Analysis in 2D

As an example, we will now consider the FEM formulation for the solution of Poisson's equation in 2D (3D can be handled similarly by using tetrahedral elements and appropriate basis functions). We have derived this previously when considering the finite difference solution, so we will only recall that the basic equation is

$$\nabla \cdot \big(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})\big) = -\rho(\mathbf{r}), \quad \mathbf{r} \in \Omega \tag{3.34}$$

with boundary conditions

$$\phi(\mathbf{r}) = \phi_D, \quad \mathbf{r} \in \Gamma_D, \tag{3.35}$$

$$\hat{n} \cdot \big(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})\big) = \kappa_N, \quad \mathbf{r} \in \Gamma_N. \tag{3.36}$$

Here, we have the entire problem domain excluding the boundaries denoted by Ω. The boundary of Ω (sometimes denoted as $\partial\Omega$ in the literature, but we will use Γ here) is subdivided into regions with Dirichlet and Neumann boundary conditions, denoted by Γ_D and Γ_N, respectively. For this problem, we have that $\Gamma_D \cup \Gamma_N = \Gamma$; i.e., the union of the Dirichlet and Neumann boundaries are equal to the complete boundary of Ω.

We can now go about formulating the FEM solution to this problem. Our first step will be to derive our weak form of the PDE. To do this, we will test (3.34) with a testing function w. This gives us

$$\int_\Omega w\big[\nabla \cdot \big(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})\big)\big] d\Omega = -\int_\Omega w\rho d\Omega. \tag{3.37}$$

We can now use integration by parts and Gauss' theorem to simplify our weak-form PDE. In particular, we can note that

$$w\big[\nabla \cdot \big(\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})\big)\big] = \nabla \cdot \big(w\epsilon\nabla\phi\big) - \epsilon\nabla w \cdot \nabla\phi \tag{3.38}$$

and

$$\int_\Omega \nabla \cdot \big(w\epsilon\nabla\phi\big) d\Omega = \oint_\Gamma \hat{n} \cdot \big(w\epsilon\nabla\phi\big) d\Gamma. \tag{3.39}$$

Using these results, we can rewrite (3.37) as

$$\int_\Omega \epsilon\nabla w \cdot \nabla\phi d\Omega - \oint_\Gamma \hat{n} \cdot \big(w\epsilon\nabla\phi\big) d\Gamma = \int_\Omega w\rho d\Omega. \tag{3.40}$$

With an eye toward using our boundary conditions, we can further rewrite this as

$$\int_\Omega \epsilon\nabla w \cdot \nabla\phi d\Omega - \oint_{\Gamma_D} \hat{n} \cdot \big(w\epsilon\nabla\phi\big) d\Gamma - \oint_{\Gamma_N} \hat{n} \cdot \big(w\epsilon\nabla\phi\big) d\Gamma = \int_\Omega w\rho d\Omega \tag{3.41}$$

by separating the integral over the entire boundary into its Dirichlet and Neumann components. We can then use the Neumann data provided in (3.36) to finally arrive at our weak-form representation of the PDE as

$$\int_\Omega \epsilon\nabla w \cdot \nabla\phi d\Omega - \oint_{\Gamma_D} \hat{n} \cdot \big(w\epsilon\nabla\phi\big) d\Gamma - \oint_{\Gamma_N} w\kappa_N d\Gamma = \int_\Omega w\rho d\Omega. \tag{3.42}$$

The next main step of the FEM process is to discretize the simulation region. Here, we will use a set of triangular elements similar to those shown in Fig. 3.4. Although not strictly necessary, it is typical that we will assume that the permittivity is constant within a particular finite element. We will then expand the scalar potential ϕ in terms of the pyramidal basis functions we developed in Section 3.5 and denoted by N_j where j is the global node number of the basis function. We will then break up the set of all nodes $\mathcal{N}$ into non-overlapping sets that correspond to all nodes on the Dirichlet boundaries $\mathcal{N}_D$ and all remaining nodes $\mathcal{N}_E$. Then, we can write ϕ as

$$\phi = \sum_{j\in\mathcal{N}_E} a_j N_j + \sum_{j\in\mathcal{N}_D} \phi_j^D N_j, \tag{3.43}$$

where ϕ_j^D is the Dirichlet data provided by (3.35) sampled at the node j.

Next, we will choose our testing functions. Due to the symmetry of the weak-form PDE given in (3.42), it appears that the Galerkin method will again be a good choice to guide our discretization. Hence, we will use as testing functions the set of functions defined by $N_j, \forall j \in \mathcal{N}_E$. Considering this, we see that the boundary integral in (3.42) over Γ_D will be 0 since all N_j with $j \in \mathcal{N}_E$ are 0 on Γ_D due to the properties of the linear interpolating functions used.

Considering this, we can plug our basis function expansion given in (3.43) into the weak-form PDE given in (3.42) and test it at a particular testing function N_i. The resulting equation is

$$\sum_{j\in\mathcal{N}_E} a_j \int_\Omega \epsilon\nabla N_i\cdot\nabla N_j d\Omega = \int_\Omega N_i\rho d\Omega + \int_{\Gamma_N} N_i\kappa_N d\Gamma - \sum_{j\in\mathcal{N}_D} \phi_j^D \int_\Omega \epsilon\nabla N_i\cdot\nabla N_j d\Omega. \tag{3.44}$$

We can repeat this process for all the different testing functions to get the matrix equation

$$[\mathbf{K}]\{\mathbf{a}\} = \{\mathbf{b}\}, \tag{3.45}$$

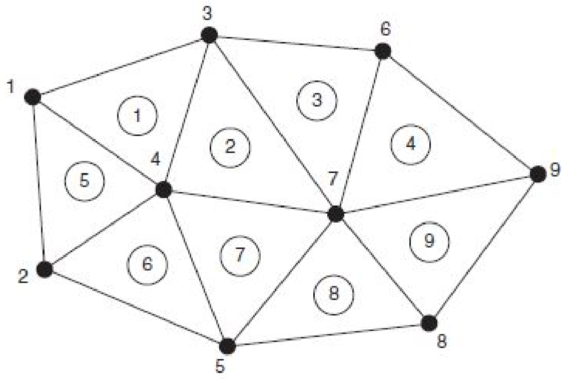
where

$$[\mathbf{K}]_{ij} = \int_\Omega \epsilon\nabla N_i\cdot\nabla N_j d\Omega, \tag{3.46}$$

$$\{\mathbf{b}\}_i = \int_\Omega N_i\rho d\Omega + \int_{\Gamma_N} N_i\kappa_N d\Gamma - \sum_{j\in\mathcal{N}_D} \phi_j^D \int_\Omega \epsilon\nabla N_i\cdot\nabla N_j d\Omega. \tag{3.47}$$

As with the previous FEM analysis we discussed, the matrix $[\mathbf{K}]$ is extremely sparse due to the small, finite support of each basis and testing function. Further, due to the simplicity of the interpolating functions used, we can either use analytical or numerical formulas to evaluate all of the integrals in (3.46) and (3.47).

Although these integrals can be readily evaluated, if we go about performing them in a naive manner we can make the computer implementation of this approach more complicated and slower. Instead, when performing an FEM analysis, it is much more common to follow what is known as an *assembly* process for calculating the matrix and excitation vectors of (3.45). The basic idea of assembly is that it can be rather difficult and inefficient to try and fully evaluate a matrix element $[\mathbf{K}]_{ij}$ due to the complicated structure of the pyramidal basis functions (e.g., depending on the mesh some elements will not overlap at all or others may overlap over multiple triangular subdomains).



(a)

e	n(1, e)	n(2, e)	n(3, e)
1	1	4	3
2	4	7	3
3	3	7	6
4	7	9	6
5	2	4	1
6	2	5	4
7	5	7	4
8	5	8	7
9	7	8	9

(b)

Figure 3.7: (a) Example triangular mesh and (b) one possible connectivity list. A counter-clockwise ordering of local nodes is used to ensure the formulas in [5] give the correct basis function values (images from [5]).

To make this process easier to code and faster, we instead go element-by-element through the mesh evaluating the integrals of the different interpolating functions defined over a particular triangular subdomain. These integrals will look like

$$K_{lk}^{(e)} = \int_{\Omega^{(e)}} \epsilon \nabla N_l^{(e)} \cdot \nabla N_k^{(e)} d\Omega \tag{3.48}$$

for a particular element e and *local node numbers* l and k. Note that the $\Omega^{(e)}$ denotes that this integration is only taken over the single triangular element where these particular linear interpolating functions are defined. We can then take these results for *all* combinations of l and k (where each index runs from 1 to 3) and add them to the different matrix elements $[\mathbf{K}]_{ij}$ that they contribute to.

To facilitate this assembly process, we generate and store what is known as a *connectivity array* or *connectivity list* during the discretization process. For a triangular mesh, this will tell us for each element what the global node numbers are that correspond to the first, second, and third local node numbers. This mapping between global and local node numbers is not unique; however, meshing tools will often automatically follow a particular convention. For example, the local node numbers always increase in a counter-clockwise order around an element as in Fig. 3.6. It is important to check these conventions when using a meshing tool to ensure that the node numbering follows any conventions you may have assumed in your calculations of basis functions. As an example, if a convention is violated it is possible for a basis function to be negative when it should be positive or point in the wrong direction if it is a vector function. An example of a simple triangular mesh and one possible connectivity list for the mesh is shown in Fig. 3.7.

We can now use the mesh and connectivity given in Fig. 3.7 to demonstrate the assembly process for a few elements. In particular, we will consider some of the terms arising from the elements 1 and 5, in that order, and will actually accumulate the results as would happen in a computer program.

1. Element 1 assembly

(a) We go to $e = 1$ in Fig. 3.7(b) and see that $l = 1$ and $k = 1$ will correspond to global node number 1 and so $K_{11}^{(1)}$ will contribute to $[\mathbf{K}]_{11}$.

(b) We then go to $l = 1$ and $k = 2$ and see that $K_{12}^{(1)}$ will contribute to $[\mathbf{K}]_{14}$.

(c) We continue this process to eventually partially fill the following matrix elements as

$$[\mathbf{K}]_{11} = K_{11}^{(1)}, \quad [\mathbf{K}]_{13} = K_{13}^{(1)}, \quad [\mathbf{K}]_{14} = K_{12}^{(1)}. \tag{3.49}$$

2. Element 5 assembly

(a) We go to $e = 5$ in Fig. 3.7(b) and see that $l = 1$ and $k = 1$ will correspond to global node number 2 and so $K_{11}^{(5)}$ will contribute to $[\mathbf{K}]_{22}$.

(b) Eventually, we can go to $l = 3$ and $k = 3$ and see these correspond to global node number 1 and so $K_{33}^{(5)}$ will contribute to $[\mathbf{K}]_{11}$. Considering we already have a value in $[\mathbf{K}]_{11}$ from element 1 assembly, we will now have that $[\mathbf{K}]_{11} = K_{11}^{(1)} + K_{33}^{(5)}$.

(c) We continue this process to eventually partially fill the following matrix elements (accumulating from the previous element 1 assembly) as

$$[\mathbf{K}]_{11} = K_{11}^{(1)} + K_{33}^{(5)}, \quad [\mathbf{K}]_{12} = K_{31}^{(5)}, \quad [\mathbf{K}]_{13} = K_{13}^{(1)}, \quad [\mathbf{K}]_{14} = K_{12}^{(1)} + K_{32}^{(5)}. \tag{3.50}$$

This process only illustrated some partial steps of the overall assembly process. More details can be found in [5, Sec. 9.2.2].

3.7 FEM Analysis of Homogeneous Waveguides

We will now look at an example of how FEM can be used in analyzing the properties of arbitrarily-shaped but homogeneously-filled waveguides. To begin, we will recall the formulation of the PDE that governs the determination of the waveguide modes and their corresponding propagation constants. To determine these general equations, we will assume that we have a geometry that is oriented along the z-axis, is infinitely long, has a constant cross sectional shape over the entirety of the waveguide, and is homogeneously-filled (i.e., ϵ and μ do not vary with $\mathbf{r}$). Under this assumption, we can assume that we will have a simple propagating wave characteristic for the z-dependence of the electric and magnetic fields contained in the waveguide. Due to the orientation of the geometry, it makes sense to break our electric and magnetic fields into their *transverse* (i.e., in the cross section of the waveguide) and *longitudinal* (i.e., along the length of the waveguide) components. Considering this, we can write our fields as

$$\mathbf{E}(x, y, z) = [\mathbf{E}_t(x, y) + \hat{z}E_z(x, y)]e^{-j\beta z}, \tag{3.51}$$

$$\mathbf{H}(x, y, z) = [\mathbf{H}_t(x, y) + \hat{z}H_z(x, y)]e^{-j\beta z}, \tag{3.52}$$

where $\mathbf{E}_t$ and $\mathbf{H}_t$ contain the transverse components (i.e., x- and y-components) of the electric and magnetic fields, respectively.

For a general waveguide analysis, we will be considering the electromagnetic fields that exist in the source-free region contained inside of a particular waveguide geometry. Hence, we can use the source-free form of Maxwell's equations to derive the Helmholtz wave equation for $\mathbf{E}$ and $\mathbf{H}$ following the standard process. This gives us for the electric field

$$\nabla^2\mathbf{E} + k^2\mathbf{E} = 0, \tag{3.53}$$

where $k = \omega\sqrt{\mu\epsilon}$ is, as usual, the wavenumber. If we write

$$\nabla^2 = \nabla_t^2 + \partial_z^2, \tag{3.54}$$

we can simplify our wave equation given the known z-dependence of our field given in (3.51). In particular, we will get that

$$\nabla_t^2\mathbf{E} + (k^2 - \beta^2)\mathbf{E} = 0. \tag{3.55}$$

At this point, it is advantageous to define a new kind of wavenumber suggested by (3.55) as

$$k_c^2 = k^2 - \beta^2. \tag{3.56}$$

We refer to k_c as the *cutoff wavenumber.* The reason for this terminology can be seen by rearranging this into the form of a *dispersion relation* as

$$\beta^2 = k^2 - k_c^2. \tag{3.57}$$

We see that we will only have wave propagation when $k^2 > k_c^2$ so that the propagation constant will not be purely imaginary.

Now, the typical strategy for actually solving practical waveguide problems is to eliminate redundant field components from Maxwell's equations given an assumed solution of the form (3.51) or (3.52). For a homogeneous waveguide, this process shows us that we can compute all field components if we know E_z and H_z. This process further shows us that E_z and H_z are *independent* of each other, so that we can break our solutions into two families known as the *transverse electric (TE)* and *transverse magnetic (TM) modes.* For this decomposition, we have that TE modes are characterized by $E_z = 0$, $H_z \neq 0$, and TM modes are characterized by $E_z \neq 0$, $H_z = 0$.

Due to the similarity between these two cases, we will only discuss the analysis of TM modes in class. From our prior discussion, we know that we only need to calculate E_z to quickly find the remaining transverse field components from this single scalar component. Hence, it will serve us well to find the equation for E_z. Here, we will have

$$\nabla_t^2\mathbf{E} + k_c^2\mathbf{E} = 0, \tag{3.58}$$

where the cutoff wavenumber is still given by $k_c^2 = k^2 - \beta^2$. In Cartesian coordinates, it is quite easy for us to simplify this to only consider the E_z components. This will give us the equation

$$\nabla_t^2 E_z + k_c^2 E_z = (\partial_x^2 + \partial_y^2)E_z + k_c^2 E_z = 0. \tag{3.59}$$

We must solve this equation subject to the boundary conditions of a particular waveguide geometry. If we are considering a simple scenario of a waveguide made from PEC, then we will have a homogeneous Dirichlet boundary condition for E_z, i.e.,

$$E_z = 0, \quad \text{on } \Gamma. \tag{3.60}$$

We can now go about formulating our FEM solution to this problem. One of the main differences from what we have discussed previously is that for (3.59), the unknown quantities that we need to solve for are both E_z and k_c^2. In a more mathematical language, we refer to this type of problem as an *eigenvalue problem.* If you recall from linear algebra, an eigenvalue λ_m and eigenvector $\{\mathbf{v}_m\}$ of a matrix $[\mathbf{A}]$ satisfy the relation

$$[\mathbf{A}]\{\mathbf{v}_m\} = \lambda_m\{\mathbf{v}_m\}. \tag{3.61}$$

Comparing this to (3.59), we see that (3.59) has the same structure where ∇_t^2 (our differential operator) is taking the role of the matrix operator in (3.61), E_z is the eigenvector, and k_c^2 is the eigenvalue.

Considering this, we can now go about formulating a weak form of the eigenvalue problem given in (3.58). This will follow the same process as we have used previously, and will give us

$$\int_\Omega \nabla_t w_i \cdot \nabla_t E_z d\Omega = k_c^2 \int_\Omega w_i E_z d\Omega, \tag{3.62}$$

where w_i is the testing function and we have used the homogeneous Dirichlet boundary condition to eliminate the integral over the boundary due to the integration by parts. The symmetry of our weak-form eigenvalue problem suggests that using the Galerkin procedure will be advantageous here. Since we are dealing with a scalar field component, we can use the pyramidal function associated with mesh nodes as our basis and testing functions. Due to the homogeneous Dirichlet boundary condition, we will only use these basis functions at nodes that do not lie on the PEC boundary of our problem.

For this formulation, our resulting matrix equation becomes

$$[\mathbf{A}]\{E_{z,m}\} = k_{c,m}^2[\mathbf{B}]\{E_{z,m}\}, \tag{3.63}$$

where m is indexing different eigenvalue and eigenvector pairs and

$$[\mathbf{A}]_{ij} = \int_\Omega \nabla_t N_i \cdot \nabla_t N_j d\Omega, \tag{3.64}$$

$$[\mathbf{B}]_{ij} = \int_\Omega N_i N_j d\Omega. \tag{3.65}$$

Comparing (3.63) to (3.61), we see that the structure of the equation is slightly different because we also have a matrix operating on the right-hand side of the equation where the eigenvalue is located. This kind of mathematical problem is known as a *generalized eigenvalue problem.* There are a number of standard numerical linear algebra techniques that can be used to solve this problem, yielding the set of eigenvalues and eigenvectors. For particularly large problems, these algorithms can take prohibitive computation times to compute all of the eigenvalues and eigenvectors. In these situations, it can sometimes be sufficient to compute only a small number of the most dominant eigenvalues and eigenvectors. There are also standard numerical linear algebra techniques to perform this computation.

As mentioned previously, we can also use this approach to analyze the TE modes of the system. The resulting PDE we need to solve has an identical form to (3.59), but involves H_z instead of E_z. The other difference is that the boundary condition must be adjusted. In particular, a homogeneous Neumann boundary condition of

$$\partial_n H_z = 0, \quad \text{on } \Gamma, \tag{3.66}$$

where ∂_n denotes a normal derivative.

3.8 Vector Basis Functions for FEM Analysis

As with the scalar formulations of FEM that we considered previously, it will be useful for us to first consider what kind of basis functions will be useful for solving vector electromagnetic problems. Typically, we will be trying to solve the vector wave equation, e.g.,

$$\nabla \times \mu_r^{-1} \nabla \times \mathbf{E} - k_0^2 \epsilon_r \mathbf{E} = 0, \tag{3.67}$$

with an additional set of appropriate Dirichlet, Neumann, or Robin boundary conditions. We can imagine formulating a weak-form solution to this equation, in which we can transfer one of the curl operations from the **E** onto the testing function. The remaining spatial derivatives that are still applied to **E** then suggest that we will need to ensure our basis functions at least have a piecewise linear variation in a manner similar to what was needed when we considered the scalar FEM formulations.

The next question that naturally arises is what component of our mesh we should have our basis functions associated with. In the scalar FEM analysis, we saw that we could develop linear interpolating functions with advantageous properties by associating a new basis function with each *node* of the mesh. If we try and do something similar for a vector function, we will quickly find that we run into difficulties when we need to ensure certain boundary conditions (such as the continuity of tangential fields across different media) are satisfied throughout our simulation domain. This problem is essentially identical to what we faced when we originally discussed trying to solve an equation like (3.67) using finite difference discretizations prior to the introduction of Yee's method. Along these same lines, the solution to our current issue is to instead represent the **E** as being associated with an *edge* of the mesh, rather than a node. This provides us with a very natural representation of **E** that automatically ensures essential properties such as the continuity of the tangential component of the field are satisfied everywhere in our solution domain.

The particular basis function that is very popular for vector-valued FEM analysis is often referred to as an *edge element* or sometimes an *edge basis*. The underlying linear variation of this basis function can be expressed in terms of the nodal linear interpolating functions

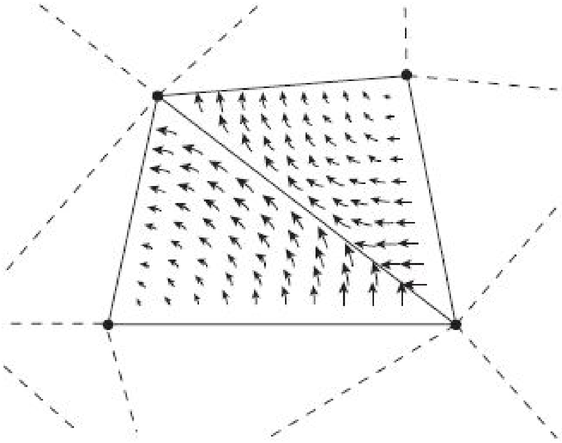


Figure 3.8: Example plot of an edge element over the triangles that share the edge of a simple triangular mesh (image from [5]).

that we used in the scalar FEM formulations previously. Recall, that we could express this pyramidal function within a particular element e for the first local node as

$$N_1^{(e)} = \begin{vmatrix} 1 & 1 & 1 \\ x & x_2 & x_3 \\ y & y_2 & y_3 \end{vmatrix} \Big/ \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}. \tag{3.68}$$

We could also easily express the functions for the remaining nodes like

$$N_2^{(e)} = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x & x_3 \\ y_1 & y & y_3 \end{vmatrix} \Big/ \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}, \tag{3.69}$$

and similar for $N_3^{(e)}$.

We can now use these functions to define the edge element for the edge of element e that runs between nodes l and k of the element. In particular, we have

$$\mathbf{N}_{lk}^{(e)}(\mathbf{r}) = \left[N_l^{(e)}\nabla N_k^{(e)} - N_k^{(e)}\nabla N_l^{(e)}\right]\ell_{lk}^{(e)}, \quad l < k. \tag{3.70}$$

In (3.70), $\ell_{lk}^{(e)}$ is a *signed length* of the edge that connects nodes l and k of element e. The particular sign that $\ell_{lk}^{(e)}$ should take is based off of a particular convention that must be followed consistently within a particular implementation of an FEM code. For example, one can establish that $\ell_{lk}^{(e)}$ takes a positive sign when the global node number of node l is less than the global node number of node k (otherwise it takes a negative sign). This kind of convention is necessary to ensure that when the basis function definition is used in elements that share a particular edge the basis function establishes a consistent vector direction along the shared edge. An illustration of the edge element for a simple triangular mesh is shown in Fig. 3.8.

From Fig. 3.8, we clearly see that the basis function will have the desired tangential continuity across the edge it is associated with. We further see that the tangential component of the basis function at all other edges of the element is 0. This is necessary to ensure the interpolating property of the basis function (similar to the pyramid functions being 0 at all adjacent nodes of the mesh). Although the *tangential component* is 0 at all the other edges of the element, the *normal component* of the basis function is not 0 across these other edges. This is important to allow the overall set of basis functions to be able to also produce discontinuous normal components of the electric field across some edges. This is important between elements that have different material properties so that the normal component of the electric field can be discontinuous, as required by the boundary condition on $\mathbf{D}$. Overall, these properties ensure that this basis function can do a sufficient job at representing $\mathbf{E}$ without causing any "pathological" problems in our discretization.

In a manner similar to the nodal elements we considered earlier, we can expand the total electric field within a particular element of the mesh through a superposition of all the edge elements for a particular triangle (for a 2D mesh) or tetrahedron (for a 3D mesh). We also will need to consider our overall basis function as a composition of the different functions defined only over a particular element given in (3.70). We will denote this overall basis function as $\mathbf{N}_j$, where j is the global edge number that the basis function is associated with. This function will take on non-zero values only over the triangles or tetrahedrons that share this particular edge. As a result, this basis function will also give us a FEM matrix that is highly sparse.

3.9 Vector FEM Analysis

With a suitable basis function developed, we now turn our attention to determining the FEM formulation for the wave equation. We will assume that there is an impressed current source within the simulation domain that can act as a source of electromagnetic fields. Under this situation, we can combine Maxwell's equations to give us the vector wave equation as

$$\nabla \times \mu_r^{-1} \nabla \times \mathbf{E} - k_0^2 \epsilon_r \mathbf{E} = -j k_0 \eta_0 \mathbf{J}_{\text{imp}}, \tag{3.71}$$

where $\eta_0 = \sqrt{\mu_0/\epsilon_0}$ is the intrinsic impedance of free space and $\mathbf{J}_{\text{imp}}$ is the impressed current source. As with other situations, to fully specify our PDE we must also consider a set of boundary conditions. To keep our formulation general, we will assume that we have both a Dirichlet and Neumann boundary conditions for this problem. These will be specified as

$$\hat{n} \times \mathbf{E} = \mathbf{P}, \quad \text{on } \Gamma_D, \tag{3.72}$$

$$\hat{n} \times \mu_r^{-1} \nabla \times \mathbf{E} = \mathbf{K}, \quad \text{on } \Gamma_N, \tag{3.73}$$

where Γ_D (Γ_N) is the surface where the Dirichlet (Neumann) boundary conditions are specified for. As with the scalar analysis case, we assume that the union of Γ_D and Γ_N cover all surfaces that need a boundary condition specified at for a particular problem. Similarly, we will denote the total simulation domain we are considering as Ω.

To develop our FEM formulation, we must first find the weak form of the wave equation given in (3.71). To do this, we take the inner product of this equation with a testing function

$\mathbf{W}$ to get

$$\int_\Omega \mathbf{W}\cdot\Big[\nabla\times\mu_r^{-1}\nabla\times\mathbf{E}-k_0^2\epsilon_r\mathbf{E}\Big]d\Omega=-jk_0\eta_0\int_\Omega\mathbf{W}\cdot\mathbf{J}_{\text{imp}}d\Omega. \tag{3.74}$$

We now want to integrate by parts to transfer one of the spatial derivatives from $\mathbf{E}$ onto the testing function $\mathbf{W}$. We can do this by noting that

$$\nabla\cdot\Big[\mathbf{W}\times\mu_r^{-1}\nabla\times\mathbf{E}\Big]=\mu_r^{-1}\big(\nabla\times\mathbf{W}\big)\cdot\big(\nabla\times\mathbf{E}\big)-\mathbf{W}\cdot\nabla\times\mu_r^{-1}\nabla\times\mathbf{E}, \tag{3.75}$$

so that (3.74) becomes

$$\int_\Omega\Big[\mu_r^{-1}\big(\nabla\times\mathbf{W}\big)\cdot\big(\nabla\times\mathbf{E}\big)-\mathbf{W}\cdot k_0^2\epsilon_r\mathbf{E}\Big]d\Omega=\\ \int_\Omega\nabla\cdot\Big[\mathbf{W}\times\mu_r^{-1}\nabla\times\mathbf{E}\Big]d\Omega-jk_0\eta_0\int_\Omega\mathbf{W}\cdot\mathbf{J}_{\text{imp}}d\Omega. \tag{3.76}$$

We can then use Gauss' theorem to rewrite the first term on the right-hand side of (3.76) as a surface integral over the boundaries of the simulation domain, giving us

$$\int_\Omega\Big[\mu_r^{-1}\big(\nabla\times\mathbf{W}\big)\cdot\big(\nabla\times\mathbf{E}\big)-\mathbf{W}\cdot k_0^2\epsilon_r\mathbf{E}\Big]d\Omega=\\ \oint_\Gamma\hat{n}\cdot\Big[\mathbf{W}\times\mu_r^{-1}\nabla\times\mathbf{E}\Big]d\Gamma-jk_0\eta_0\int_\Omega\mathbf{W}\cdot\mathbf{J}_{\text{imp}}d\Omega. \tag{3.77}$$

We can then break the surface integral up into its pieces over the Dirichlet and Neumann surfaces to get

$$\int_\Omega\Big[\mu_r^{-1}\big(\nabla\times\mathbf{W}\big)\cdot\big(\nabla\times\mathbf{E}\big)-\mathbf{W}\cdot k_0^2\epsilon_r\mathbf{E}\Big]d\Omega=\int_{\Gamma_D}\mu_r^{-1}\big(\hat{n}\times\mathbf{W}\big)\cdot\big(\nabla\times\mathbf{E}\big)d\Gamma\\ -\int_{\Gamma_N}\mathbf{W}\cdot\mathbf{K}d\Gamma-jk_0\eta_0\int_\Omega\mathbf{W}\cdot\mathbf{J}_{\text{imp}}d\Omega. \tag{3.78}$$

In transitioning from (3.77) to (3.78), we have used simple vector algebraic identities to reorder cross products and the scalar triple products. We have also substituted in the result of our Neumann boundary condition from (3.73). Overall, we can identify (3.78) as our weak form of the vector wave equation.

With the weak form in hand, we are now ready to choose the basis and testing functions. As mentioned when we discussed the edge element, we see that this basis function will be able to do a good job representing $\mathbf{E}$ for this weak form since we have transferred one of the spatial derivatives from $\mathbf{E}$ to $\mathbf{W}$. We also see that there is a relatively strong symmetry in the weak form of the vector wave equation, which suggests to us that using the Galerkin method for choosing our testing function will lead to a good numerical system. In this case, the Galerkin's method results in us selecting to use edge elements for the testing function.

Considering this, we can express $\mathbf{E}$ as

$$\mathbf{E} = \sum_{j \in \mathcal{E}_E} a_j \mathbf{N}_j + \sum_{j \in \mathcal{E}_D} E_j^D \mathbf{N}_j, \tag{3.79}$$

where we have separated the entire set of mesh edges $\mathcal{E}$ into two non-overlapping sets (i.e., they are disjoint) that correspond to all edges on the Dirichlet boundaries $\mathcal{E}_D$ and all other edges $\mathcal{E}_E$. We can determine the values of E_j^D from the Dirichlet boundary condition data provided in (3.72).

We can now substitute (3.79) into the weak-form PDE (3.78) and test it with a particular testing function $\mathbf{N}_i$. The resulting equation is

$$\begin{aligned}\sum_{j \in \mathcal{E}_E} a_j \int_\Omega \Big[\mu_r^{-1}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) - k_0^2 \epsilon_r \mathbf{N}_i \cdot \mathbf{N}_j\Big] d\Omega = -\int_{\Gamma_N} \mathbf{N}_i \cdot \mathbf{K} d\Gamma \\ - jk_0\eta_0 \int_\Omega \mathbf{N}_i \cdot \mathbf{J}_{\text{imp}} d\Omega - \sum_{j \in \mathcal{E}_D} E_j^D \int_\Omega \Big[\mu_r^{-1}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) - k_0^2 \epsilon_r \mathbf{N}_i \cdot \mathbf{N}_j\Big] d\Omega.\end{aligned} \tag{3.80}$$

Note that the additional integral over Γ_D that is present in (3.78) does not appear because the testing functions are taken from the set $\mathcal{E}_E$, which all have zero tangential component along Γ_D due to the interpolating properties of the edge elements.

We can repeat this process for all the different testing functions to get the matrix equation

$$[\mathbf{K}]\{\mathbf{a}\} = \{\mathbf{b}\}, \tag{3.81}$$

where

$$[\mathbf{K}]_{ij} = \int_\Omega \Big[\mu_r^{-1}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) - k_0^2 \epsilon_r \mathbf{N}_i \cdot \mathbf{N}_j\Big] d\Omega, \tag{3.82}$$

$$\begin{aligned}\{\mathbf{b}\}_i = -\int_{\Gamma_N} \mathbf{N}_i \cdot \mathbf{K} d\Gamma - jk_0\eta_0 \int_\Omega \mathbf{N}_i \cdot \mathbf{J}_{\text{imp}} d\Omega \\ - \sum_{j \in \mathcal{E}_D} E_j^D \int_\Omega \Big[\mu_r^{-1}(\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) - k_0^2 \epsilon_r \mathbf{N}_i \cdot \mathbf{N}_j\Big] d\Omega.\end{aligned} \tag{3.83}$$

As with the previous FEM analysis we discussed, the matrix $[\mathbf{K}]$ is extremely sparse due to the small, finite support of each basis and testing function. Further, due to the simplicity of the edge elements used, we can either use analytical formulas to evaluate all of the integrals in (3.82) and (3.83) for the common case where it is assumed that the material properties take constant values over a single element of the mesh. Alternatively, we can use numerical integration methods.

As with the scalar FEM analysis, it is necessary to follow an element-by-element assembly process to evaluate the different parts that contribute to the overall entries in the matrix and excitation vector. More details and a simple example of this assembly process can be found in [5].

3.10 FEM Analysis of Inhomogeneous Waveguides

We will now consider how to analyze arbitrarily shaped waveguides that are inhomogeneously filled. To calculate the eigenvalues and eigenmodes of such a general waveguide, the vector wave equation must be used. By combining Maxwell's curl equations, and noting that the relative permeability and permittivity are space dependent, the wave equation in an inhomogeneous, source-free medium, denoted as Ω, is found to be

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times \mathbf{E} \right) - k_0^2 \epsilon_r \mathbf{E} = 0, \tag{3.84}$$

with the homogeneous Dirichlet boundary condition

$$\hat{n} \times \mathbf{E} = 0 \text{ on } \Gamma. \tag{3.85}$$

The weak form of (3.84) can be found by following a process similar to what we already considered previously in class and using the homogeneous Dirichlet boundary condition to simplify the results from integration by parts. This leads to the desired general weak-form equation,

$$\int_\Omega \left\{ \frac{1}{\mu_r} \big(\nabla \times \mathbf{W}_i \big) \cdot \big(\nabla \times \mathbf{E} \big) - k_0^2 \epsilon_r \mathbf{W}_i \cdot \mathbf{E} \right\} d\Omega = 0. \tag{3.86}$$

With a few exceptions where certain symmetries exist, a general inhomogeneous waveguide is not able to support the usual TE and TM modes of homogeneous waveguides. This is a result of the phase matching condition only being able to be met if both E_z and H_z are present in the waveguide. Instead, hybrid modes are found to exist, denoted as EH and HE modes depending on whether the mode reduces to a TM or TE mode at cutoff, respectively.

The general characteristic of a propagating mode in a longitudinally-uniform waveguide is a simple complex exponential dependence for the variable in the longitudinal direction, assumed here as the z-direction. This allows the general form of the electric field in the waveguide to then be written compactly as

$$\mathbf{E}(x, y, z) = \left[\frac{1}{\beta} \mathbf{E}_t(x, y) + j \hat{z} E_z(x, y) \right] e^{-j\beta z}, \tag{3.87}$$

where β is the propagation constant, $\mathbf{E}_t$ is the transverse component of the field, and E_z is the longitudinal component of the field. By choosing the testing function to be the complex conjugate of this, the weak-form equation can be simplified. Note that although we describe this as choosing the testing function to be the "complex conjugate" of the basis function, we are actually still using Galerkin's method. The underlying reason is that when we take the inner product of the wave equation with the testing function to find the weak form of the PDE, we should generally be taking the complex conjugate of the testing function to be using a well-defined inner product (recall our discussion around the L^2 function space). We have not had to worry about this up to this point because we had always being using real-valued basis and testing functions. Since that is no longer the case, we now need to take these complex conjugates into account correctly to arrive at a well-performing weak form of the PDE. Considering this, the exact expression that we will substitute into (3.86) is

$$\mathbf{W}(x,y,z) = \left[\frac{1}{\beta}\mathbf{W}_t(x,y) - j\hat{z}W_z(x,y)\right]e^{j\beta z}. \tag{3.88}$$

We can substitute these two expressions for the electric field and testing function into (3.86) to find the weak form of the PDE for the inhomogeneous waveguide problem. To find this, it will be useful for us to recall that

$$\begin{aligned}\nabla\times\mathbf{E} &= \nabla\times\left[\frac{1}{\beta}\mathbf{E}_t(x,y) + j\hat{z}E_z(x,y)\right]e^{-j\beta z}\\ &= \left[\frac{1}{\beta}\nabla_t\times\mathbf{E}_t - j\hat{z}\times\nabla_t E_z - j\hat{z}\times\mathbf{E}_t\right]e^{-j\beta z}.\end{aligned} \tag{3.89}$$

Using a similar expression for $\nabla\times\mathbf{W}$ and a set of standard vector algebraic identities, we can simplify the weak-form expression of our PDE to be

$$\begin{aligned}\int_\Omega\Bigg\{\frac{1}{\mu_r}\big(\nabla_t\times\mathbf{W}_t\big)\cdot\big(\nabla_t\times\mathbf{E}_t\big) - k_0^2\epsilon_r\mathbf{W}_t\cdot\mathbf{E}_t&\\ +\beta^2\left[\frac{1}{\mu_r}\big(\mathbf{W}_t+\nabla_t W_z\big)\cdot\big(\mathbf{E}_t+\nabla_t E_z\big) - k_0^2\epsilon_r W_z E_z\right]\Bigg\}\,d\Omega = 0.&\end{aligned} \tag{3.90}$$

At this point, we need to choose the particular basis functions we will use in our FEM formulation. Since we have both vector and scalar quantities that we need to expand (i.e., $\mathbf{E}_t$ and E_z), we will need to use two sets of functions. The particularly sensible choice is to discretize the longitudinal components (which are scalars) with the nodal basis function that we used earlier for the homogeneous waveguide problem. We can then expand the vector transverse fields using the standard edge element that we discussed previously. Using these functions, we have

$$\mathbf{E}_t = \sum_{j=1}^{N_{edge}}\mathbf{N}_j E_{t,j} \tag{3.91}$$

and

$$E_z = \sum_{j=1}^{N} N_j E_{z,j} \tag{3.92}$$

where N_{edge} is the number of edge elements and N is the number of nodes. In both cases, the edges and nodes which reside on the conducting boundary are not counted due to the homogeneous Dirichlet boundary condition assumed in our problem formulation. By substituting these expressions into (3.90) and defining the testing functions similar to (3.91) and (3.92) the full FEM system matrix can be compactly written as

$$\begin{bmatrix}A_{tt} & 0\\ 0 & 0\end{bmatrix}\begin{Bmatrix}E_t\\ E_z\end{Bmatrix} = -\beta^2\begin{bmatrix}B_{tt} & B_{tz}\\ B_{zt} & B_{zz}\end{bmatrix}\begin{Bmatrix}E_t\\ E_z\end{Bmatrix}, \tag{3.93}$$

where

$$[A_{tt}]_{ij} = \int_\Omega \left[\frac{1}{\mu_r} \big(\nabla_t \times \mathbf{N}_i\big) \cdot \big(\nabla_t \times \mathbf{N}_j\big) - k_0^2 \epsilon_r \mathbf{N}_i \cdot \mathbf{N}_j \right] d\Omega \tag{3.94}$$

$$[B_{tt}]_{ij} = \int_\Omega \frac{1}{\mu_r} \mathbf{N}_i \cdot \mathbf{N}_j \, d\Omega \tag{3.95}$$

$$[B_{tz}]_{ij} = \int_\Omega \frac{1}{\mu_r} \mathbf{N}_i \cdot \nabla_t N_j \, d\Omega \tag{3.96}$$

$$[B_{zt}]_{ij} = \int_\Omega \frac{1}{\mu_r} \nabla_t N_i \cdot \mathbf{N}_j \, d\Omega \tag{3.97}$$

$$[B_{zz}]_{ij} = \int_\Omega \left[\frac{1}{\mu_r} \nabla_t N_i \cdot \nabla_t N_j - k_0^2 \epsilon_r N_i N_j \right] d\Omega. \tag{3.98}$$

From (3.93), we see that we have arrived at a generalized eigenvalue problem that may be solved for a set of eigenvalues, β^2, and eigenvectors, $\{E_t\}$ and $\{E_z\}$.

From (3.93), we also see that we can eliminate the nodal basis functions from our generalized eigenvalue problem if desired. This can be done by using the lower half of the system matrix in (3.93) to find a relationship between the node- and edge-based basis functions. From this, it is seen that

$$[B_{zt}]\{E_t\} + [B_{zz}]\{E_z\} = 0, \tag{3.99}$$

so that

$$\{E_z\} = -[B_{zz}]^{-1}[B_{zt}]\{E_t\}. \tag{3.100}$$

Substituting this into the matrix equation defined by the upper half of the system matrix it is seen that the generalized eigenvalue problem may be expressed as

$$[A_{tt}]\{E_t\} = -\beta^2 \big([B_{tt}] - [B_{tz}][B_{zz}]^{-1}[B_{zt}]\big)\{E_t\}, \tag{3.101}$$

which may then be solved for β and $\{E_t\}$. If $\{E_z\}$ is required for further analysis it may be easily computed from (3.100).

Although it is easy to rewrite the expressions in this way, there are some important numerical aspects to consider regarding whether this is advantageous or not, particularly in the case when the number of basis functions are large. The central issue is computing the inverse of $[B_{zz}]$. As mentioned previously, the computational complexity of this operation can scale as $O(N^3)$ for a general matrix, where N would be the number of nodes basis functions are defined at in this case. If the system size is large, this may be an impractical operation to perform.

A further issue with trying to directly compute the inverse relates to matrix storage. The matrix $[B_{zz}]$ is highly sparse due to the nature of our FEM solution approach. As a result, $[B_{zz}]$ can be stored very efficiently, with $O(N)$ memory complexity where a normal full matrix would require $O(N^2)$ memory complexity. The issue is that even though $[B_{zz}]$ is highly sparse, the inverse will typically be completely dense. As a result, the memory complexity of directly computing the inverse also has the potential to greatly increase the memory usage of this method. One possible way to try and avoid this is using a standard numerical linear algebra technique to compute a *sparse approximate inverse* or use other specialized linear algebra methods for inverting sparse matrices. These can help keep the computational cost and memory complexity controlled, but will invariably result in some approximation errors in the method.

Another possible option is to consider the expression (3.101) to be more "formal" than literal, i.e., we are writing this expression to suggest a particular way to make the expressions compact but we have no intention of literally computing the matrix inverse. This kind of expression can happen frequently in the literature to help illustrate more clearly what the steps of an algorithm may be practically equivalent to, even though we may take a more efficient route to implement it. For instance, for many numerical linear algebra techniques that can solve the generalized eigenvalue problem of (3.101), they do not actually need to explicitly store the matrix, they only need to be able to have access to a computer routine that can return the result of a matrix-vector product between a supplied vector and the matrix we are interested in. For the sequence of operations on the right-hand side of (3.101), we could first compute the matrix-vector product of $[B_{zt}]\{E_t\}$. We can then use an iterative solver to compute the effect of $[B_{zz}]^{-1}$ in a much lower computational complexity than a general-purpose direct solver would require due to the sparsity of $[B_{zz}]$. After using the iterative solver, we can continue evaluating the remaining matrix-vector products to provide the overall effect of the right-hand side of (3.101) to the generalized eigenvalue problem solver.

3.10.1 Numerical Results

To illustrate the utility and validity of the developed method, we present a few numerical results computed using the numerical algorithm discussed in Section 3.10. The first case tested was the calculation of the dispersion curves for the first few modes of an empty rectangular waveguide (these may also be computed using the method described in Section 3.7). From an elementary analysis, the complete set of waveguide modes may be easily calculated analytically. Upon doing this, the change in the propagation constant as a function of frequency may be calculated to be

$$k_z = \sqrt{k^2 - \left(\frac{m\pi}{a}\right)^2 - \left(\frac{n\pi}{b}\right)^2}, \tag{3.102}$$

where k_z is the propagation constant, k is the wavenumber of the homogeneous media filling the waveguide, and a and b are the width and height of the waveguide, respectively. The integer constants m and n specify the mode, and represent the number of half-wavelength variations along the respective dimensions of the waveguide.

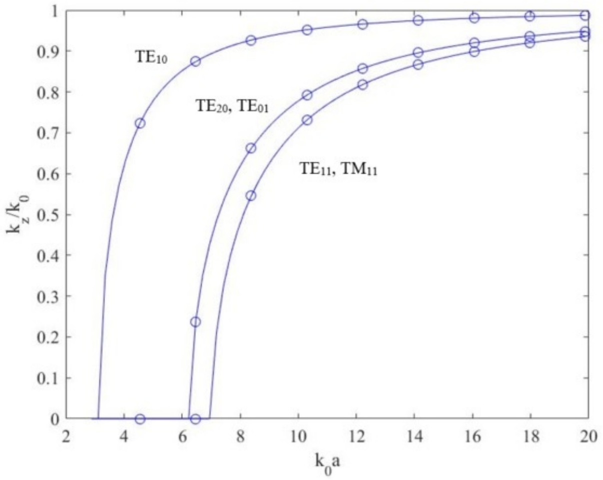


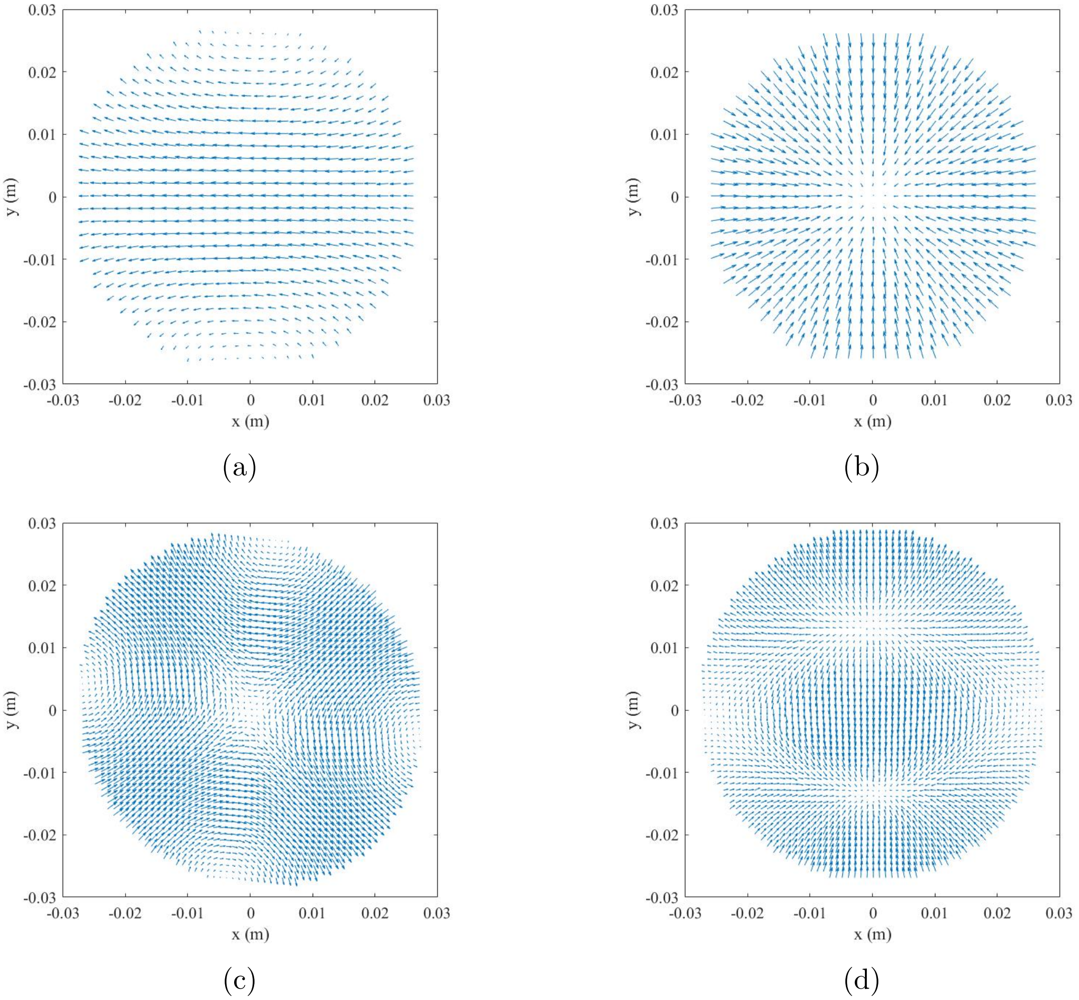
Figure 3.9: Comparison of analytical (solid lines) and FEM (circles) calculated dispersion curves of first 5 modes in a rectangular waveguide with $b = a/2$.

The dispersion curves for the first 5 modes of a hollow rectangular waveguide are shown in Fig. 3.9. The waveguide dimensions are set so that $b = a/2$. The solid lines in this figure are the analytical result, while the circles are the propagation constant calculated by the FEM code. Clearly, excellent agreement is achieved between the two.

The next problem solved is the calculation of the modes in a circular waveguide. Instead of plotting the dispersion curves in a manner similar to Fig. 3.9, the FEM code is used to calculate the eigenvectors. Vector plots of these modes are then produced by using the interpolation properties of the edge basis functions. Due to the plotting function used, the z-component of the electric field is not being plotted. As can be seen in Figs. 3.10(a) to 3.10(d), the interpolation is able to reproduce the smoothly varying fields well. These plots may be compared to those in books such as [5] or [19], and are seen to agree well in general structure.

The final numerical example presented is the calculation of the dispersion curves in an inhomogeneously-filled waveguide. In general, analytical solutions do not exist for these types of geometry, however, if certain symmetries exist with respect to the waveguide structure and the inhomogeneity, analytical results are still possible. To validate the FEM solution for this type of problem, a geometry in which analytical solutions are still possible was considered. The particular structure studied is a rectangular waveguide which is half-filled along the vertical direction with a dielectric material with relative permittivity of 4.

Finding the analytical result for this structure is more complicated, with only highlights of the process reproduced here. Full details on the derivation may be found in [5]. The key difference between the inhomogeneously-filled waveguide and a homogeneously-filled one is that TE and TM modes are no longer able to be supported. Rather, both E_z and H_z fields must be present at the same time, leading to what are known as hybrid modes. Despite this complication, the analysis is still similar to that used in simpler electromagnetics problems. The general approach is to determine viable expressions for the electromagnetic fields in both of the homogeneous regions, and then enforce the various continuity and phase



(a)

(b)

(c)

(d)

Figure 3.10: Electric field distribution of low order modes in a hollow circular waveguide. Particular modes are (a) TE_{11}, (b) TM_{01}, (c) TE_{21}, and (d) TM_{11}.

matching conditions along the interface. Once this is done a matrix equation may be formed to determine the coefficients of the longitudinal fields in both regions. Finding a non-trivial solution to this problem yields two transcendental equations,

$$\frac{\mu_1}{k_{1y}} \tan k_{1y} h = -\frac{\mu_2}{k_{2y}} \tan k_{2y}(b-h) \tag{3.103}$$

$$\frac{k_{1y}}{\epsilon_1} \tan k_{1y} h = -\frac{k_{2y}}{\epsilon_2} \tan k_{2y}(b-h) \tag{3.104}$$

which may be solved separately for k_z, the propagation constant, with the constraints that

$$k_1^2 = \omega^2 \mu_1 \epsilon_1 = k_x^2 + k_{1y}^2 + k_z^2 \tag{3.105}$$

$$k_2^2 = \omega^2 \mu_2 \epsilon_2 = k_x^2 + k_{2y}^2 + k_z^2. \tag{3.106}$$

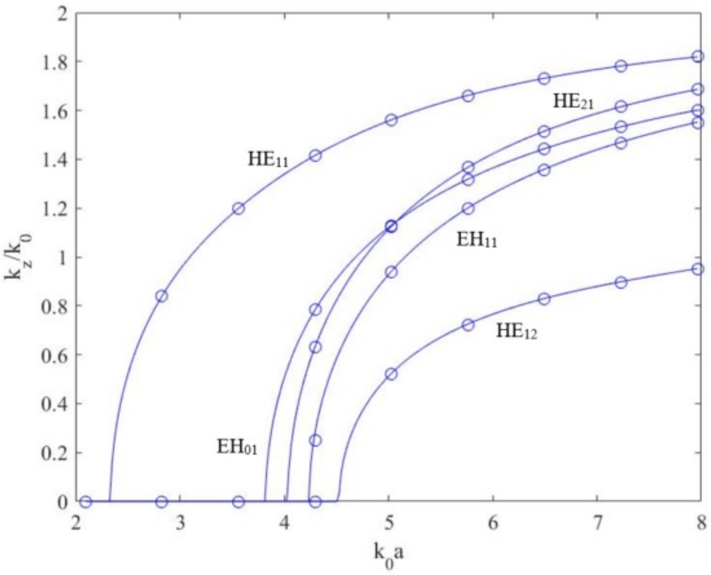


Figure 3.11: Comparison of the analytical (solid lines) and FEM (circles) calculated propagation constants of the inhomogeneously-filled rectangular waveguide with $b = a/2$, with one half having material with $\epsilon_r = 4$ and the other half with $\epsilon_r = 1$.

In these equations, a subscript of 1 or 2 denotes the region the specified parameter is associated with. The k_{1y} and k_{2y} are the wavenumbers in the y-direction in each region, k_x is the wavenumber in the x-direction (which is the same for both regions due to phase matching), and h is the height of the first region. The naming convention for the hybrid modes is dictated by the behavior that they exhibit at cutoff, that is, whether the E_z or H_z component dominates. With a little further analysis, it is seen that (3.103) correspond to EH modes, which reduce to TM modes at cutoff (so that E_z is non-zero). Consequently, (3.104) correspond to HE modes, which reduce to TE modes at cutoff.

The FEM solution of the propagation constant is compared to the analytical results in Fig. 3.11. As can be seen in Fig. 3.11, an excellent match is achieved between the analytical and FEM calculated results. One interesting point to note is that the dispersion curves occasionally intersect, indicating when the modes are degenerate. This can cause complications in calculating dispersion curves numerically with an eigenmode solver because we can only calculate the entire set of eigenvalues and eigenvectors at discrete frequency points. As a result, correctly identifying which mode a particular eigenvalue should be associated with after moving away from a degeneracy point becomes important so we can draw a correct interpolating curve between the different data points. This problem is often referred to as *mode tracking*, and appears frequently in various fields of engineering and physics that frequently use modal decompositions. Various techniques exist to try and automate the problem of properly tracking modes as a function of frequency. One simple approach is to compute and store the eigenvectors for all modes of interest at two different frequency points. These modes can be numerically integrated against one another (i.e., we are taking their inner product). Due to the orthogonality of the different eigenvectors, we will typically only get an appreciable result from this numerical integration if the two eigenmodes correspond to the same eigenmode, just with slightly different eigenvalues due to the change in frequency.

3.11 Open Regions in FEM Analysis

Just as with the finite difference methods, there are situations where we want to analyze “open” regions using FEM, such as computing the performance of antennas or solving various scattering problems. To keep the solution region finite sized, we will need to devise different kinds of approaches to “terminate” the solution region in the form of boundary conditions that are suitable for incorporation in our FEM formulations. We will consider a few different cases in class, but there are a number of other approaches that also exist [5, 20]. As we will see shortly, similar methods to what we used for finite difference methods will also be applicable to FEM formulations. However, the different solution approach used in FEM analysis complicates the implementation of a number of these approaches, making matching the performance of finite difference methods sometimes difficult to achieve. Despite this difficulty, sufficient performance is still certainly possible with FEM formulations so that accurate solutions can be achieved for almost any practical application.

3.11.1 Waveguide Ports

The first kind of terminating boundary we will consider for “open” problems is relevant to waveguide analysis. If we want to compute the scattering parameters of some kind of transmission line device, it is necessary for us to be able to “terminate” a port with a matched load to ensure that no reflections are produced. The equivalent view is that we want to replicate the effect of having our port actually extend to infinity with a constant cross section matching that of the transmission line at the port so that the wave can continue on indefinitely without producing any reflections. Due to the difficulty of defining impedances with arbitrary waveguides, we cannot typically rely on something as simple as just placing some kind of “resistive load” at the port to approximately absorb the incident waves. Instead, we will need an approximate boundary condition that “absorbs” the fields associated with a particular waveguide mode with minimal reflections.

To derive such a boundary condition, we will need to know the field distribution and associated propagation constant of the relevant waveguide modes that we intend to consider in our simulation. For simple geometries, such as homogeneously-filled rectangular cross sections, we can compute these mode distributions and propagation constants analytically. However, in general, this will not be the case so that we will need to instead compute these properties using a different method, such as the FEM formulations for analyzing waveguide modes that we discussed previously in class.

We will now consider a relatively simple case to derive our approximate boundary condition. In particular, we will assume that we have a single port that we are performing our analysis at and that the port is located far enough away from the “device” being analyzed that at the location of our port (where we will need to derive our boundary condition) only a single dominant mode of the waveguide is non-negligible. The analysis of more complicated cases involving multiple modes can be found in [5, Ch. 9.3.3]. On the more practical side, we can help ensure these approximations are closer to reality for an actual analysis by placing our port at the end of a section of constant cross-section waveguide or transmission line that we explicitly model in our simulation region. The goal is to have this cross section be long enough that the field distribution along the cross section approaches that of the ideal infinitely-long waveguide or transmission line geometry so that it can be absorbed effectively by the approximate boundary condition we will now derive.

Now, given the assumptions mentioned above, we have that the total electric field at the port region can be expressed as

$$\begin{aligned}\mathbf{E}(u,v,w) &= \mathbf{E}^{\text{inc}}(u,v,w) + \mathbf{E}^{\text{ref}}(u,v,w) \\ &= E_0\mathbf{e}_T(u,v)e^{-j\beta w} + \Gamma E_0\mathbf{e}_T(u,v)e^{j\beta w},\end{aligned} \tag{3.107}$$

where u, v, and w define a local coordinate system at the port region with u and v denoting the transverse dimensions and w pointing in the longitudinal direction toward the simulation region. Further, we have that E_0 is the complex-valued amplitude of the incident field, $\mathbf{e}_T$ is the electric field modal distribution in the transverse plane and β is the propagation constant of this mode. If we were dealing with the dominant mode of a simple rectangular waveguide then we would have

$$\mathbf{e}_T(u,v) = \hat{v}\sin\left(\frac{\pi u}{a}\right), \tag{3.108}$$

$$\beta = \sqrt{k^2 - \left(\frac{\pi}{a}\right)^2}, \tag{3.109}$$

where a is the width of the waveguide along the u dimension.

We can derive our approximate boundary condition following a similar process to what we did to derive ABCs when we studied the FDTD method. There we took the derivative of the field solution to determine a kind of Robin boundary condition that involved the field and its derivative. Here, we can take the tangential portions of the curl of (3.107) to find that

$$\hat{n} \times (\nabla \times \mathbf{E}) = -j\beta\mathbf{E}^{\text{inc}} + j\beta\mathbf{E}^{\text{ref}}, \tag{3.110}$$

where $\hat{n}$ is the unit normal of the port aperture, and will thus point in the $\pm w$-direction. Now, because the reflected field is not known *a priori* it is desirable to eliminate it from this expression. We can do this by using (3.107) again so that we arrive at

$$\hat{n} \times (\nabla \times \mathbf{E}) = j\beta\mathbf{E} - 2j\beta\mathbf{E}^{\text{inc}}. \tag{3.111}$$

We can rewrite this into a more standard form for use as a boundary condition as

$$\hat{n} \times (\nabla \times \mathbf{E}) + j\beta\hat{n} \times (\hat{n} \times \mathbf{E}) = -2j\beta\mathbf{E}^{\text{inc}}. \tag{3.112}$$

This boundary condition can be readily incorporated into the weak-form solution of the wave equation within a waveguide geometry. If we need to consider the case where a device has multiple ports, we need to apply a similar boundary condition at the other ports to act as a "matched load" for calculations of the scattering parameters of the device. Since these ports are not excited with an incident field for the scattering parameter extraction, we can simply use a homogeneous version of the above boundary condition; i.e.,

$$\hat{n} \times (\nabla \times \mathbf{E}) + j\beta\hat{n} \times (\hat{n} \times \mathbf{E}) = 0 \tag{3.113}$$

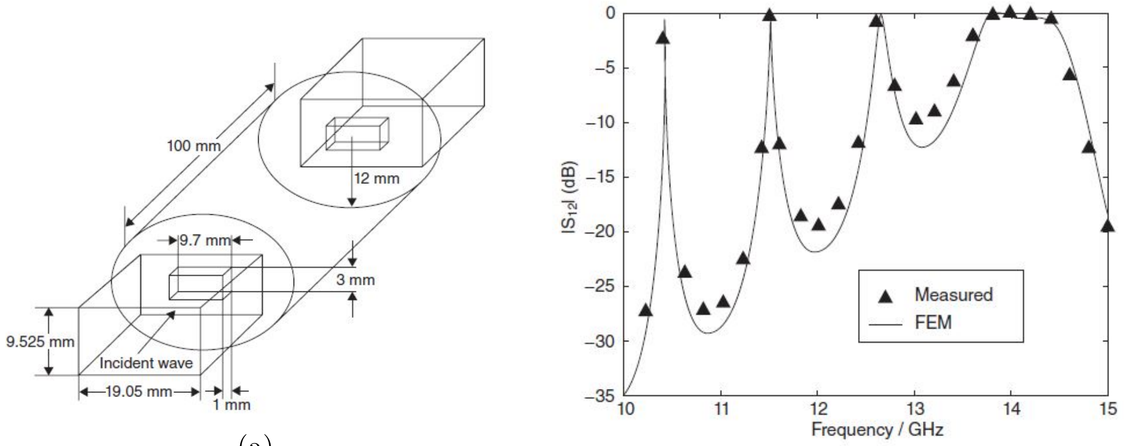
to terminate all other ports of the device.

To allow this port to be placed closer to the device being analyzed, higher order modes can be included in the derivation of the boundary condition as well. A similar analysis can also be used to allow for the situation where a single physical aperture can excite multiple propagating modes. More details on this kind of extension can be found in [5].

A final note is that this kind of boundary condition can be found in commercially available CEM tools such as CST and HFSS. In these tools, this boundary condition and/or excitation source for the model is typically referred to as a "wave port". Our discussion above about needing to have some section of constant cross section of the transmission line or waveguide explicitly modeled within the simulation region is typically necessary to achieve accurate results. Further, these tools often apply a PEC boundary condition along the outer extent of the wave port when they solve the 2D FEM problem to compute the modes of the waveguide or transmission line structure. As a result, when analyzing "open" kinds of transmission lines, such as microstrip lines, it is important to make the cross section of the wave port large enough that these artificial PEC boundaries do not significantly alter the solved for field distribution from that of an ideal "open" transmission line.

To demonstrate the utility of this method we will look at the results from two simulations. The first involves the analysis of a cylindrical cavity resonator that is coupled to from two rectangular apertures. These apertures are then excited by rectangular waveguides that have wave ports placed at the ends of them to facilitate the computation of the scattering parameters of the device. The cavity geometry and results are shown in Fig. 3.12. The numerical results are compared to the measured scattering parameters, where very good agreement is seen.

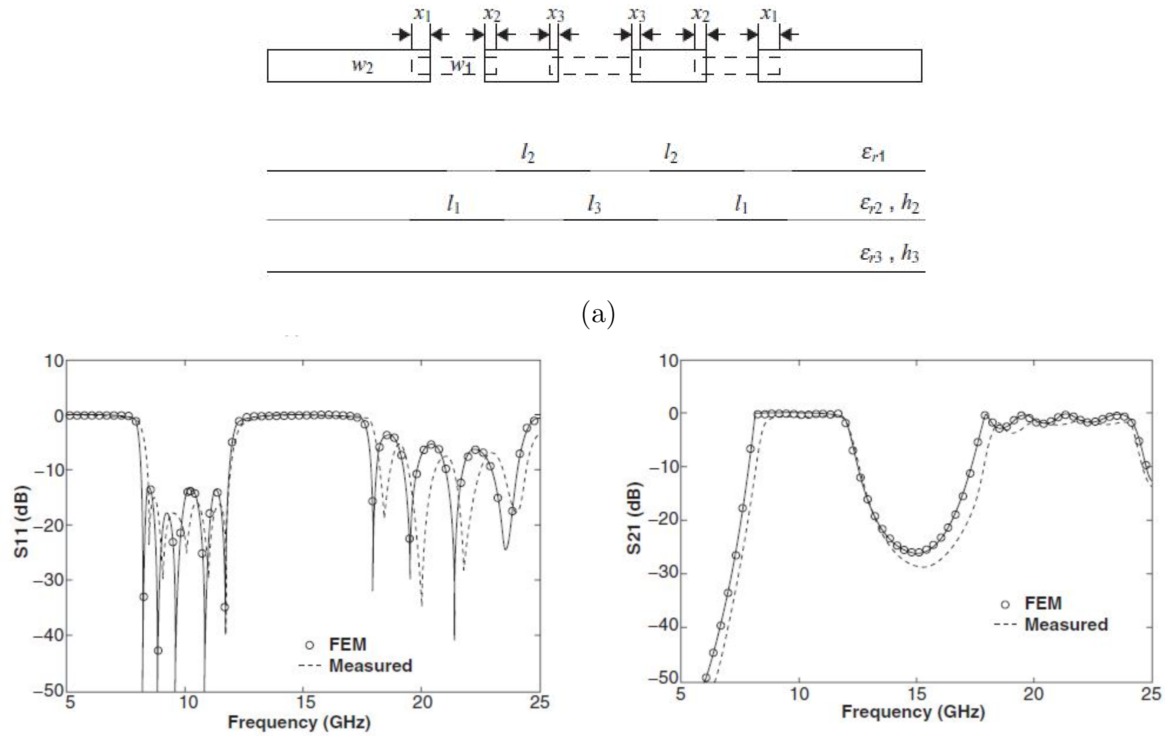
The next example considers the analysis of a microstrip filter. The filter is implemented on a two-layer circuit board with coupling between different sections of the filter achieved through a (predominantly) capacitive interaction between overlapping microstrip traces on different layers of the board. The geometry of the filter and corresponding numerical and measured results are shown in Fig. 3.13. Generally good agreement is seen between the numerical and measured results, however, not nearly as good as for the previous example. The most likely reason for this is that the precision of fabrication and assembly of the printed circuit board can be less than a waveguide geometry, and that the circuit boards involve multiple dielectric materials which likely have permittivity values that deviate from their design values (sometimes to a fairly significant degree for certain materials). In particular, it appears that many of the FEM resonances predicted occur at lower frequencies than for the measured device, which suggests that the permittivity values used in the simulation were larger than what the actual physical implementation had. This kind of deviation is very common if additional steps in the design process are not taken to properly characterize the materials and fabrication processes being utilized.



(a)

(b)

Figure 3.12: Use of wave ports to analyze a cylindrical cavity resonator. (a) The geometry analyzed and (b) the comparison of numerical and measured results (images from [21]).



(a)

(b)

(c)

Figure 3.13: Use of wave ports to analyze a microstrip filter. (a) The geometry analyzed, (b) the $|S_{11}|$, and (c) the $|S_{21}|$ (images from [22]).

3.11.2 Absorbing Boundary Conditions

We may follow a similar process to what was done with wave ports to see how to utilize an absorbing boundary condition (ABC) for use in analyzing "open" region problems with FEM. Recall from our discussions about the FDTD method that we could develop an ABC by taking the spatial derivative of an assumed outward propagating wave. If we assume that we were far enough away from the scattering or radiating object then the outward propagating wave will approximately look like a plane wave. When we perform the differentiation, we arrive at an ABC in a general region of the form (assuming the ABC is in free space)

$$\hat{n} \times (\nabla \times \mathbf{E}^{\text{sc}}) + jk_0 \hat{n} \times (\hat{n} \times \mathbf{E}^{\text{sc}}) = 0, \tag{3.114}$$

which is also referred to as the *Sommerfeld radiation condition.* This can serve as a first-order ABC for our FEM analysis.

To use this boundary condition, we recall that the decomposition of the total field into its incident and scattered components allows us to write $\mathbf{E}^{\text{sc}} = \mathbf{E} - \mathbf{E}^{\text{inc}}$. We can substitute this into our first-order ABC of (3.114) to get the boundary condition

$$\hat{n} \times (\nabla \times \mathbf{E}) + jk_0 \hat{n} \times (\hat{n} \times \mathbf{E}) = \mathbf{U}^{\text{inc}}, \tag{3.115}$$

where

$$\mathbf{U}^{\text{inc}} = \hat{n} \times (\nabla \times \mathbf{E}^{\text{inc}}) + jk_0 \hat{n} \times (\hat{n} \times \mathbf{E}^{\text{inc}}). \tag{3.116}$$

Since the incident field is known *a priori* over the artificial closing surface of our simulation region that the ABC is being applied to, we can easily compute the value of (3.116) over the ABC surface. The remainder of the ABC in (3.115) is a *Robin boundary condition* that can be incorporated into the FEM solution following standard manipulations.

Although this process is quite simple, you should recall from our discussion around ABCs for the FDTD method that the performance of a first-order ABC is fairly limited unless the scattered field very much resembles a plane wave with an almost normal incidence angle. This requires placing the ABC a far distance from the object being analyzed, increasing the size of the matrix equation that must be solved in the FEM analysis. As a result, it is desirable to improve performance of the ABC significantly so that the computation region can be minimized.

One way to do this is in FEM analysis is to make the artificial boundary that the ABC will be applied to curved. This curvature can be used to lower the amount of "empty space" that needs to be simulated, and can also better match the field characteristics of the scattered fields closer to the object being analyzed. Although this appears promising, this approach is hampered by a few different factors when attempting to derive higher-order ABCs in 3D. To see these issues, we will briefly look at a few steps in the derivation process of a higher-order ABC. In 3D, this involves taking the appropriate sequence of spatial derivatives of the asymptotic expansion of a vector wave solution given by

$$\mathbf{E}(r, \theta, \phi) = \frac{e^{-jkr}}{r} \sum_{n=0}^{\infty} \frac{\mathbf{A}_n(\theta, \phi)}{r^n}. \tag{3.117}$$

There is a degree of freedom in defining these derivatives that is usually taken as an arbitrary parameter s that can be selected to try and optimize the ABC for a particular purpose. When this derivation approach is used to derive a first-order ABC, the result is

$$\hat{r} \times \nabla \times \mathbf{E} + jk\hat{r} \times \hat{r} \times \mathbf{E} + (s-1)\nabla_t E_r = 0. \tag{3.118}$$

It is customary to take $s = 1$ to remove the final term of this ABC, since the presence of only a single spatial derivative in this form within the FEM solution will lead to a non-symmetric matrix system that is desired to be avoided. When this is done, the ABC reduces to the Sommerfeld radiation condition that was already given in (3.114).

If we extend this derivation approach, the second-order ABC becomes

$$\begin{aligned}&\hat{r} \times \nabla \times \mathbf{E} + jk\hat{r} \times \hat{r} \times \mathbf{E} \\ &\qquad - \frac{r}{2(jkr+1)}\left[\nabla \times \big(\hat{r}\hat{r} \cdot \nabla \times \mathbf{E}\big) + (s-1)\nabla_t(\nabla_t \cdot \mathbf{E}) - (2-s)jk\nabla_t E_r\right] = 0.\end{aligned} \tag{3.119}$$

To maintain the symmetry of the FEM matrix, it is desirable to set $s = 2$. However, this compromises the "symmetry" of the performance of the ABC for TE and TM polarized waves. As a result, the ABC will then not perform equally as well for all polarizations, which is undesirable. Although this is not good, it is more important to maintain the symmetry of the FEM matrix so that it is still typical to choose $s = 2$. Despite this simplification, there are additional issues around evaluating the $\nabla_t(\nabla_t \cdot \mathbf{E})$ terms within the FEM solution. The issue is that when vector edge-based elements are used to expand $\mathbf{E}$ we cannot readily use integration by parts to transfer derivatives away from $\mathbf{E}$. This is because the surface divergence of $\mathbf{E}$ is not continuous due to the discontinuity of edge elements in the normal direction across the edges between elements. As a result, specialized approaches need to be used to accurately enforce the second-order ABC. The overall takeaway is that implementing higher-order ABCs in FEM analysis over curved surfaces involves a number of difficulties that are challenging to accommodate within an FEM formulation.

Before continuing on, it is worth briefly emphasizing why it was so desirable to keep a symmetric FEM matrix system, even if the cost was a (slight) loss in accuracy of the ABC. There are two main reasons related to matrix storage and access to iterative solvers. The matrix storage conclusion is rather obvious: if a matrix is symmetric we only need to store half of the matrix since the other half is identical. For large-scale analysis, this saving in memory can greatly improve the range of problems that can be analyzed before requiring sophisticated computer science solutions to the challenges of efficiently accessing and using data from large data structures. The second reason symmetric matrices are preferred is that it allows us to use specialized numerical algorithms that only apply to symmetric matrices. For instance, a special form of the biconjugate gradient iterative solver can be developed for the kinds of complex symmetric matrices that FEM analysis in electromagnetics produces [20]. Each iteration of this algorithm can typically be completed in a faster time than the corresponding iteration of a standard conjugate gradient method. Further, the biconjugate gradient method typically converges with fewer iterations as well. The combination of these effects can make the biconjugate gradient method five to six times faster than the conjugate gradient method, which is why it is so desirable to maintain the symmetry of the FEM system matrix [20].

3.11.3 Perfectly Matched Layers

We now turn our attention to the final termination we will consider for "open region" problems with FEM, which is perfectly matched layers (PMLs). When we discussed PMLs in the context of the FDTD method, we saw that we could derive the PML using a few different methods that in the end were equivalent from a performance perspective. The two that we focused on were the formulation of the PML using a coordinate stretching method and a related interpretation of the PML as an anisotropic absorber. Within the context of FEM formulations, the anisotropic absorber is much easier to implement with minimal change to an existing FEM code. Due to this simplicity, we will focus on the anisotropic absorber viewpoint here.

Recall from our discussions during the FDTD derivation that we could derive the anisotropic absorber viewpoint from the coordinate stretching one by finding a way to express the stretched coordinate form of Maxwell's equations in a manner that "looked" like a standard version of Maxwell's equations (essentially, we were just regrouping where the "stretching" parameters were in the equations). From this derivation, we saw that our normal form of Maxwell's equations should look like

$$\nabla \times \mathbf{E} = -j\omega\overline{\boldsymbol{\mu}} \cdot \mathbf{H}, \tag{3.120}$$

$$\nabla \times \mathbf{H} = j\omega\overline{\boldsymbol{\epsilon}} \cdot \mathbf{E}, \tag{3.121}$$

$$\nabla \cdot (\overline{\boldsymbol{\epsilon}} \cdot \mathbf{E}) = 0, \tag{3.122}$$

$$\nabla \cdot (\overline{\boldsymbol{\mu}} \cdot \mathbf{H}) = 0, \tag{3.123}$$

within a source-free region and where

$$\overline{\boldsymbol{\mu}} = \mu\overline{\boldsymbol{\Lambda}}, \tag{3.124}$$

$$\overline{\boldsymbol{\epsilon}} = \epsilon\overline{\boldsymbol{\Lambda}}, \tag{3.125}$$

$$\overline{\boldsymbol{\Lambda}} = \begin{bmatrix} \dfrac{s_y s_z}{s_x} & 0 & 0 \\ 0 & \dfrac{s_z s_x}{s_y} & 0 \\ 0 & 0 & \dfrac{s_x s_y}{s_z} \end{bmatrix}. \tag{3.126}$$

Each of the stretching parameters had a form similar to

$$s_x = 1 - j\frac{\sigma_x}{\omega\epsilon}. \tag{3.127}$$

To incorporate this version of the PML into our FEM solution we need to start by revisiting the derivation of our wave equation so that we can formulate an appropriately modified weak-form solution. To do this, we start by taking the inverse of $\overline{\boldsymbol{\mu}}$ in (3.120) to get

$$\overline{\boldsymbol{\mu}}^{-1} \cdot \nabla \times \mathbf{E} = -j\omega \mathbf{H}. \tag{3.128}$$

We can then take the curl of (3.128) and substitute in the result from (3.121) to get the following wave equation

$$\nabla \times \overline{\boldsymbol{\mu}}^{-1} \cdot \nabla \times \mathbf{E} - \omega^2 \overline{\boldsymbol{\epsilon}} \cdot \mathbf{E} = 0. \tag{3.129}$$

We can readily formulate the weak-form solution of this PDE by performing our testing process and using integration by parts. This gives us

$$\int_\Omega \Big[\big(\nabla \times \mathbf{W}\big) \cdot \overline{\boldsymbol{\mu}}^{-1} \cdot \big(\nabla \times \mathbf{E}\big) - \omega^2 \mathbf{W} \cdot \overline{\boldsymbol{\epsilon}} \cdot \mathbf{E} \Big] d\Omega = 0, \tag{3.130}$$

where we have assumed that all boundary terms from the integration by parts vanish for simplicity. In reality, these terms will need to be taken into account in the same way as they were previously for the standard vector wave equation within the region of the simulation domain of interest (where there is no PML absorber). Despite this simplification, the main takeaway is that the inclusion of the anisotropic absorber within the FEM formulation is quite straightforward. Further, it maintains its benefits of being able to have its performance systematically improved by increasing the thickness or loss of the absorber.

3.12 Finite Element Analysis in the Time Domain

The finite element method is often used to perform frequency domain analysis. However, when a very large number of frequency points need to be simulated it can become advantageous to consider the use of a time domain method. Further, it is often much easier to analyze nonlinear systems in the time domain. It is of course possible to analyze these problems using the FDTD method we have learned about previously, but this comes at the cost of losing the improved geometric and solution fidelity that is possible to achieve with FEM. As a result, it is of interest to also be able to perform finite element analysis in the time domain. These codes are often referred to as finite element time domain (FETD) solvers, and represent a particularly powerful computational technique for the analysis of complicated electromagnetic problems. We will now consider the basic details of formulating this kind of method.

To begin, we need to revisit the formulation of our weak-form PDE. Our starting point is the time domain version of Maxwell's equations, which are

$$\nabla \times \mathbf{E} = -\mu \partial_t \mathbf{H} \tag{3.131}$$

$$\nabla \times \mathbf{H} = \epsilon \partial_t \mathbf{E} + \sigma \mathbf{E} + \mathbf{J}_i. \tag{3.132}$$

To keep our discussion general, we will assume that we have the following boundary conditions with time-varying data:

$$\hat{n} \times \mathbf{E} = \mathbf{P}(t) \quad \text{on } \Gamma_D \tag{3.133}$$

and

$$\hat{n} \times \mu^{-1} \nabla \times \mathbf{E} + Y \hat{n} \times \hat{n} \times \partial_t \mathbf{E} = \mathbf{K}_N(t) \quad \text{on } \Gamma_N. \tag{3.134}$$

The boundary condition in (3.134) represents a kind of impedance boundary condition with Y being the value of the surface admittance of the boundary Γ_N.

We can follow our standard process of forming the wave equation in an inhomogeneous medium to convert (3.131) and (3.132) to

$$\nabla \times \mu^{-1} \nabla \times \mathbf{E} + \epsilon \partial_t^2 \mathbf{E} + \sigma \partial_t \mathbf{E} = -\partial_t \mathbf{J}_i. \tag{3.135}$$

We can next form our weak form of this PDE by testing (3.135) with $\mathbf{W}_i$ over the entire volume of interest, yielding

$$\int_\Omega \left[\mathbf{W}_i \cdot \nabla \times \mu^{-1} \nabla \times \mathbf{E} + \epsilon \mathbf{W}_i \cdot \partial_t^2 \mathbf{E} + \sigma \mathbf{W}_i \cdot \partial_t \mathbf{E} \right] d\Omega = - \int_\Omega \mathbf{W}_i \cdot \partial_t \mathbf{J}_i d\Omega. \tag{3.136}$$

We can perform an integration by parts on the first term in the same way that we did in the frequency domain to then get

$$\begin{aligned} \int_\Omega & \left[\mu^{-1} (\nabla \times \mathbf{W}_i) \cdot (\nabla \times \mathbf{E}) + \epsilon \mathbf{W}_i \cdot \partial_t^2 \mathbf{E} + \sigma \mathbf{W}_i \cdot \partial_t \mathbf{E} \right] \\ &= - \int_{\Gamma_N} \left[Y (\hat{n} \times \mathbf{W}_i) \cdot (\hat{n} \times \partial_t \mathbf{E}) + \mathbf{W}_i \cdot \mathbf{K}_N \right] d\Gamma_N - \int_\Omega \mathbf{W}_i \cdot \partial_t \mathbf{J}_i d\Omega \end{aligned} \tag{3.137}$$

after substituting in for our impedance boundary condition given in (3.134). Note that we have implicitly assumed in the derivation of (3.137) that our testing function will satisfy $\hat{n} \times \mathbf{W}_i = 0$ on Γ_D since we will not need testing functions at the edges of the mesh where the value of $\hat{n} \times \mathbf{E}$ is already known due to the Dirichlet boundary condition given in (3.133).

With the necessary weak-form of our PDE in hand, we can continue our FEM discretization by expanding $\mathbf{E}$ with a set of basis functions. Since we are performing a vector finite element analysis, it will be sensible to still expand the spatial variation of $\mathbf{E}$ using the same vector edge elements that we used in our frequency domain analysis. We will then assume that our expansion coefficients for these edge elements change as a function of time so that we can write

$$\mathbf{E} = \sum_{j \in \mathcal{E}_E} a_j(t) \mathbf{N}_j + \sum_{j \in \mathcal{E}_D} E_j^D(t)\, \mathbf{N}_j, \tag{3.138}$$

where we have separated the entire set of mesh edges $\mathcal{E}$ into two non-overlapping sets (i.e., they are disjoint) that correspond to all edges on the Dirichlet boundaries $\mathcal{E}_D$ and all other edges $\mathcal{E}_E$. We can determine the values of E_j^D from the Dirichlet boundary condition data provided in (3.133).

If we substitute (3.138) into (3.137) and follow the Galerkin method to choose $\mathbf{W}_i = \mathbf{N}_i$, we end up with a second-order ordinary differential equation with “matrix coefficients” as

$$[T] \frac{d^2}{dt^2} \{a\} + [R] \frac{d}{dt} \{a\} + [S]\{a\} = \{f\}, \tag{3.139}$$

where

$$[T]_{ij} = \int_\Omega \epsilon \mathbf{N}_i \cdot \mathbf{N}_j d\Omega, \tag{3.140}$$

$$[R]_{ij} = \int_\Omega \sigma \mathbf{N}_i \cdot \mathbf{N}_j d\Omega + \int_{\Gamma_N} Y(\hat{n} \times \mathbf{N}_i) \cdot (\hat{n} \times \mathbf{N}_j) d\Gamma_N, \tag{3.141}$$

$$[S]_{ij} = \int_\Omega \mu^{-1} (\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) d\Omega \tag{3.142}$$

$$\begin{aligned}\{f\}_i = -\sum_{j \in \mathcal{E}_D} \int_\Omega \left[\mu^{-1} (\nabla \times \mathbf{N}_i) \cdot (\nabla \times \mathbf{N}_j) E_j^D + \mathbf{N}_i \cdot \mathbf{N}_j \left(\epsilon \frac{d^2 E_j^D}{dt^2} + \sigma \frac{d E_j^D}{dt} \right) \right] d\Omega \\ - \int_{\Gamma_N} \mathbf{N}_i \cdot \mathbf{K}_N d\Gamma_N - \int_\Omega \mathbf{N}_i \cdot \partial_t \mathbf{J}_i d\Omega.\end{aligned} \tag{3.143}$$

There are a number of different ways that we can now go about discretizing the problem in the temporal dimension.

The simplest approaches are to use direct integration or finite difference methods as we have discussed previously in the context of FDTD. To do this, we discretize the time axis uniformly so that $t \to n\Delta t$ with $n = 0, 1, \ldots$. We can now use our standard finite difference formulas to approximate the temporal derivatives in (3.139). If we use central differences for both derivatives, we can find a time-stepping equation as

$$\begin{aligned}\left\{ \frac{1}{(\Delta t)^2}[T] + \frac{1}{2\Delta t}[R] \right\} \{a\}^{n+1} = \left\{ \frac{2}{(\Delta t)^2}[T] - [S] \right\} \{a\}^n \\ - \left\{ \frac{1}{(\Delta t)^2}[T] - \frac{1}{2\Delta t}[R] \right\} \{a\}^{n-1} + \{f\}^n.\end{aligned} \tag{3.144}$$

We can obviously use this equation in a time-marching process to advance our solution in time given proper initial conditions for $\{a\}$. The main difference between (3.144) and the time-marching methods that we dealt with when we discussed the FDTD method is that because $[T]$ and $[R]$ are non-diagonal matrices we will need to solve a matrix equation *during each time step* of our simulation.

If the matrix sizes are small enough, we can simply compute the "inverse" of the matrix equation (either explicitly or through an LU decomposition) at the beginning of the simulation and then reuse it throughout the entire time-marching process. If the matrix sizes are too large for the use of a direct solver, we can instead use an iterative solver in every time step of the simulation. To make this process efficient, we typically will want to form an effective preconditioner that we can use to improve the convergence of the iterative solver. Since the matrices do not change throughout the simulation, we only need to form this preconditioner once. This can have useful advantages compared to frequency domain methods if a very dense frequency sampling is needed because every frequency point must usually be treated independently by forming a new preconditioner or computing a new matrix "inverse".

Now, just as with the FDTD method, we could decide to use different finite differencing formulas to perform our discretization. However, using forward or backward differences will naturally reduce our accuracy to first-order whereas the central difference formula given in (3.144) is second-order accurate. Similar to FDTD, there are also implications to the *stability* of the method depending on the finite difference formulas that are used. A stability analysis shows that the forward differencing approach is unconditionally *unstable* while the backward differencing approach would be unconditionally *stable*. Again, similar to FDTD, the central difference formula is conditionally stable. However, because the FETD approach has an unstructured grid we are no longer able to derive a "simple" stability condition in the way we could for FDTD. Instead, the stability condition can only be derived in terms of the properties of the matrices involved. We will consider this in more detail shortly, but for now we will mention the approximate result that the stability condition can usually be estimated as

$$\Delta t < 0.3 h_{\min}/c \tag{3.145}$$

for first-order finite elements, where $h_{\min}$ is the size of the smallest element in the mesh [20]. This estimate can be extended to higher-order elements if $h_{\min}$ is divided by the order of the element used before plugging into (3.145).

Another popular choice for completing the temporal discretization is to use a time-marching method derived from the Newmark-beta integration method. This approach is equivalent to using central differencing for the first and second order derivatives and using a specially-designed weighted average for the undifferentiated quantities. For the situation in (3.139), we would approximate the undifferentiated quantities as

$$\{a\} \approx \beta\{a\}^{n+1} + (1-2\beta)\{a\}^n + \beta\{a\}^{n-1} \tag{3.146}$$

$$\{f\} \approx \beta\{f\}^{n+1} + (1-2\beta)\{f\}^n + \beta\{f\}^{n-1}, \tag{3.147}$$

where β is a parameter that can be selected between 0 and 1 to achieve different performance characteristics. If $\beta = 0$, we reduce back to the central difference case. The most common choice is for $\beta \geq 1/4$, which results in a method which is both unconditionally stable and second-order accurate. For this method, the time-stepping formula becomes

$$\begin{aligned}\left\{\frac{1}{(\Delta t)^2}[T] + \frac{1}{2\Delta t}[R] + \beta[S]\right\}\{a\}^{n+1} &= \left\{\frac{2}{(\Delta t)^2}[T] - (1-2\beta)[S]\right\}\{a\}^n \\ -\left\{\frac{1}{(\Delta t)^2}[T] - \frac{1}{2\Delta t}[R] + \beta[S]\right\}&\{a\}^{n-1} + \beta\{f\}^{n+1} + (1-2\beta)\{f\}^n + \beta\{f\}^{n-1}.\end{aligned} \tag{3.148}$$

Although there are a number of differences between the two time-stepping formulas given in (3.144) and (3.148), the main one of interest is that the matrix $[S]$ occurs in the left-hand side of (3.148) but doesn't in (3.144). The $[S]$ matrix is a discrete representation of the $\nabla \times \nabla \times$ operator, which typically produces a very ill-conditioned matrix (this can be viewed as being related to the large null space that exists for this operator in the continuum case). This ill-conditioning makes the convergence of an iterative solver typically take much longer for the time-stepping equation given in (3.148) compared to that of (3.144). As a result, the time-stepping formula in (3.144) will sometimes be referred to as an *explicit method* because the matrix solution can typically be completed quite easily, while the time-stepping formula in (3.148) will be referred to as an *implicit method* due to the difficulty in solving iteratively from the presence of $[S]$ [20].

3.12.1 Stability Analysis

We will now briefly consider the stability analysis of the time-stepping formulas discussed in the previous section. Since we no longer have a regular grid like in the FDTD stability analysis, we will need to follow a slightly more sophisticated analysis approach here. The particular approach we will use is a Z-transform analysis to determine the conditions on Δt that will result in a stable region of convergence. If we take the Z-transform of (3.144) with the loss set to 0 for simplicity (i.e., $[R] = 0$) and the source term set to 0, we get

$$\frac{1}{(\Delta t)^2}[T]z\{\tilde{a}\} - \left\{\frac{2}{(\Delta t)^2}[T] - [S]\right\}\{\tilde{a}\} + \frac{1}{(\Delta t)^2}[T]z^{-1}\{\tilde{a}\} = 0, \tag{3.149}$$

where $\{\tilde{a}\}$ is our expansion coefficient vector in the Z-transform domain. We can rearrange this as

$$\frac{1}{(\Delta t)^2}[T]z^2\{\tilde{a}\} - \left\{\frac{2}{(\Delta t)^2}[T] - [S]\right\}z\{\tilde{a}\} + \frac{1}{(\Delta t)^2}[T]\{\tilde{a}\} = 0 \tag{3.150}$$

and then factor the polynomial equation to get

$$-\frac{(z-1)^2}{z}\{\tilde{a}\} = (\Delta t)^2[T]^{-1}[S]\{\tilde{a}\}. \tag{3.151}$$

We can now consider this to be an eigenvalue equation for the matrix $(\Delta t)^2[T]^{-1}[S]$ with eigenvalue

$$\lambda = -(z-1)^2/z. \tag{3.152}$$

We can now inspect the region of convergence for this Z-transform analysis in terms of the eigenvalue λ. In particular, we can rearrange (3.152) as

$$(z-1)^2 + \lambda z = 0 \tag{3.153}$$

to determine for which values of λ the roots of this polynomial equation in z will be contained within the unit circle of the complex plane (and hence, denotes stability). A simple analysis shows that the max value that λ can take is 4.

We may now return to our eigenvalue equation in (3.151) and substitute in the largest allowed eigenvalue of this equation to determine a stability condition on Δt. In particular, we get that

$$4\{\tilde{a}\} = (\Delta t)^2[T]^{-1}[S]\{\tilde{a}\} \tag{3.154}$$

can be rearranged into a stability condition as

$$\boxed{\Delta t \leq \frac{2}{\sqrt{\rho\big([T]^{-1}[S]\big)}}} \tag{3.155}$$

where $\rho\big([T]^{-1}[S]\big)$ denotes the *spectral radius* of the matrix $[T]^{-1}[S]$, which corresponds to its largest eigenvalue. It should be noted that although performing a complete eigenvalue decomposition can be rather time-consuming, only finding the *largest eigenvalue* of a matrix can be performed quite efficiently. In this case, we would actually seek the largest eigenvalue of the generalized eigenvalue problem

$$[S]\{x\} = \lambda[T]\{x\} \tag{3.156}$$

to avoid computing the inverse of $[T]$.

We can repeat this analysis for the time-stepping formula from the Newmark-beta method given in (3.148). For this method, the characteristic equation becomes

$$(z-1)^2 + \lambda[\beta z^2 + (1-2\beta)z + \beta] = 0 \tag{3.157}$$

in a lossless medium. The roots of this can be found to be

$$z = \frac{2 - \lambda(1-2\beta) \pm \sqrt{(1-4\beta)\lambda^2 - 4\lambda}}{2(1+\lambda\beta)} \tag{3.158}$$

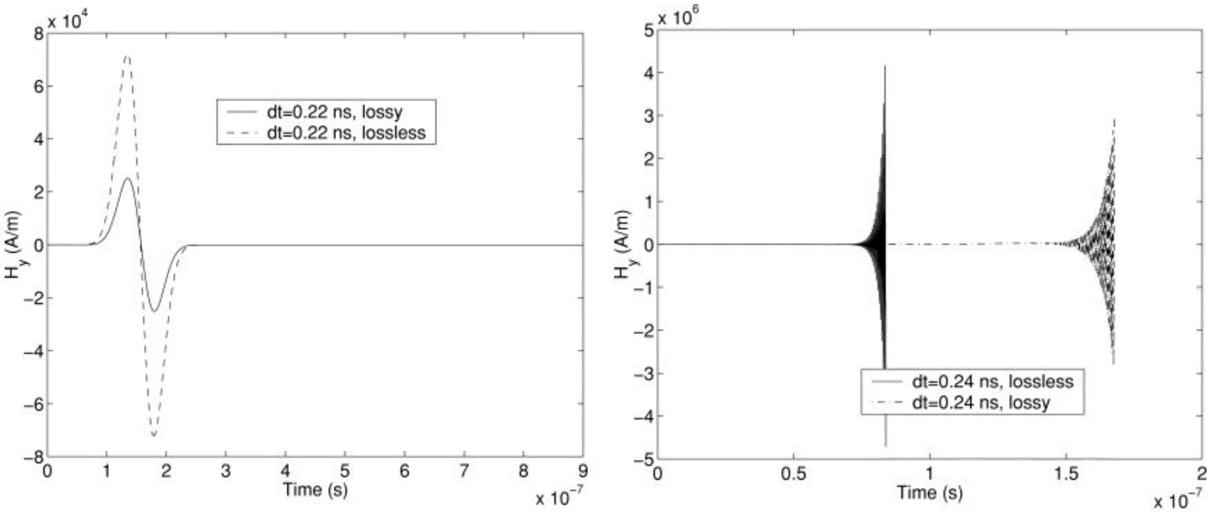
which results in roots that are always on the unit circle if $\beta \geq 1/4$ [20]. Hence, this method will be unconditionally stable for this range of β values.

The stability analysis can also be completed for the case where there is loss in the system. The end result of the analysis is that the loss *does not* impact the stability condition. This may seem counter-intuitive at first, but the issue is that no matter what conductivity is included in our analysis its loss will not be sufficient to counteract the *exponential growth* that is caused by the incorrect temporal step size with respect to the mesh being analyzed. That being said, the loss *will* still impact the time when the instability "appears" in the simulation. An illustration of this effect is shown in Fig. 3.14.

3.12.2 Numerical Results

As a simple numerical example, an FETD code was used to model the same cavity-backed microstrip patch antenna that the usual frequency domain FEM also analyzed [24]. The geometry of the patch antenna is shown in Fig. 3.15 while the numerical results for the input impedance are compared in Fig. 3.16. Clearly, there is generally good agreement between the results over the entire frequency band modeled when only a single FETD simulation is performed.

One possible reason for the discrepancy between the two methods is the different boundary condition used to terminate the open region. The frequency domain FEM results used a hybrid method that involves terminating the FEM region with a boundary integral region (which can be solved using the method of moments, another computational technique



(a)

(b)

Figure 3.14: Illustration of the effect of loss on the stability of FETD methods. (a) The results in a lossless and lossy media when the stability condition is satisfied and (b) when the stability condition is not satisfied (images from [23]). Clearly, the loss does not change the stability condition but it does result in the obvious instability appearing at a later simulation time.

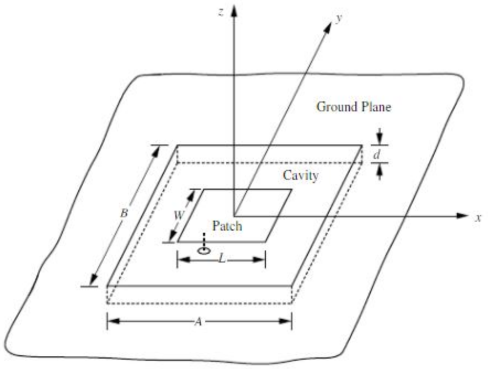
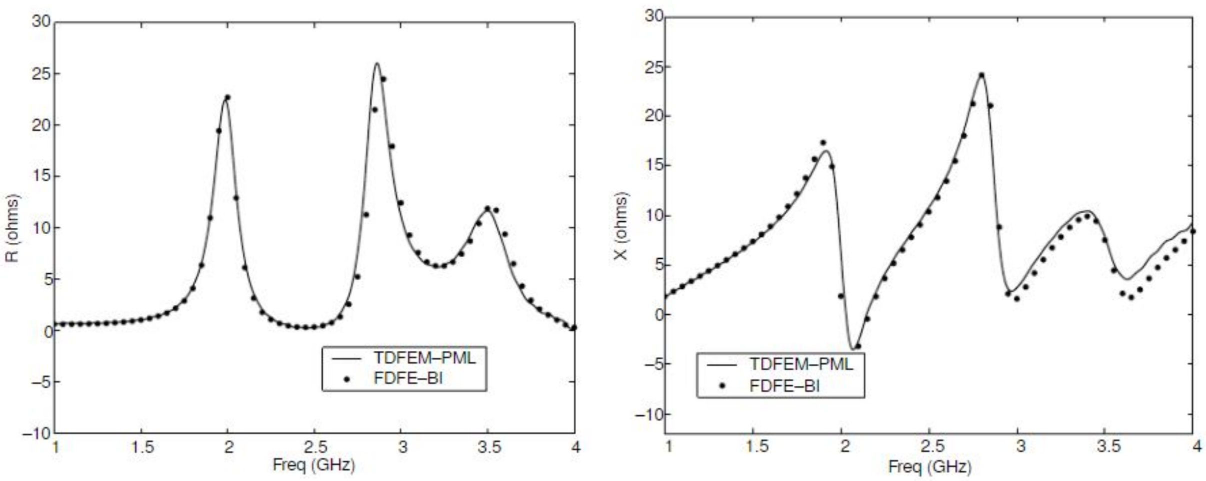


Figure 3.15: Geometry of the cavity-backed patch antenna modeled using the frequency domain FEM and FETD codes (image from [24]).

we will discuss later in the course). This kind of boundary integral termination is more computationally expensive, but it also provides a higher accuracy than what can typically be achieved with an ABC or PML termination. Meanwhile, the FETD code is terminated using a PML. It should be noted that the PML termination for the FETD code is much more involved than in the frequency domain FEM. The reason for this can be understood by recalling that for the FEM PML both the permittivity and permeability are treated as anisotropic dispersive materials. As a result, the FETD code must be developed to handle both electrically and magnetically dispersive materials simultaneously. The approximation involved in this process and the general lower accuracy of a PML compared to a boundary



(a)

(b)

Figure 3.16: Results for the (a) real and (b) imaginary part of the antenna input impedance using the FETD and FEM codes (images from [24]).

integral are likely the reasons for the discrepancy between the two results in Fig. 3.16.

3.13 Basics of Mesh Generation

One of the key pre-processing steps in FEM is to generate a suitable mesh of a potentially complex geometry to perform the desired analysis. In general, mesh generation is a field of study itself, with different meshing algorithms being developed to try and improve performance. We will not go into depth on the different kinds of meshing algorithms that exist in this course, but will instead focus on discussing certain terminology and considerations that go into evaluating whether a mesh will likely lead to suitable results for CEM analysis.

3.13.1 Types of Meshes

When classifying the *type* of mesh that is being used, there are a few terms we typically use. Of primary importance is the type of *cell* or *element* that is used in making the mesh. Up to now, we have primarily discussed triangular and tetrahedral meshes because they generally are able to represent arbitrarily curved surfaces with sufficient quality for many applications. However, one can also use different elements, such as quadrilateral or hexahedral meshes. Examples of different mesh elements that are commonly used are shown in Fig. 3.17. These different mesh elements are not used as frequently in CEM applications, but they have occasionally been used for certain applications where they are potentially more natural. It should be noted that the definitions of basis functions used in the FEM analysis will naturally need to be modified depending on the mesh cell or element used.

Another way to classify a mesh is whether it is *conforming* or not. This terminology can sometimes also be used in a different contexts related to meshing (e.g., if it is conforming to a curve or surface), but in this instance we are referring to whether the mesh has any

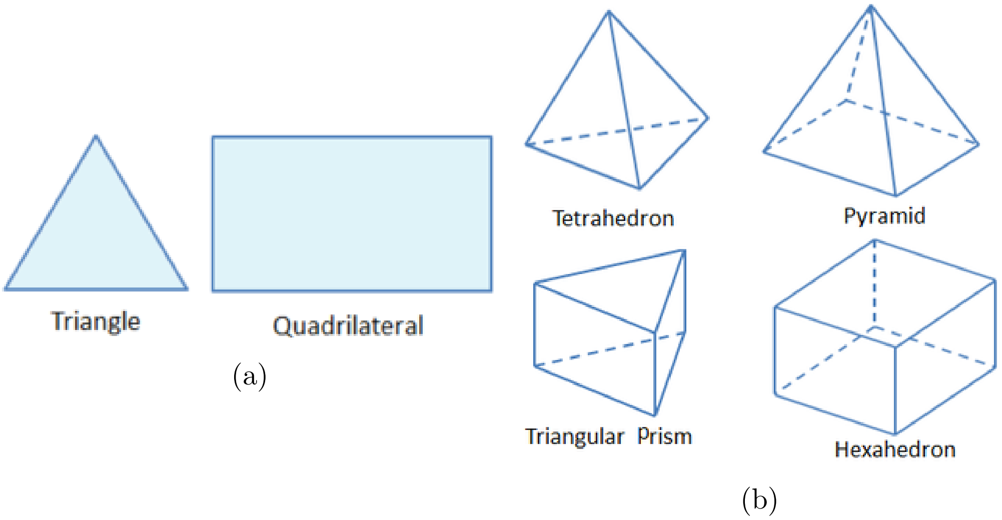


Figure 3.17: Different kinds of (a) 2D and (b) 3D mesh cells that are commonly used in computational analysis (images from [25]).

hanging nodes. A hanging node in a mesh is a node that lies along an edge of a particular element, but that is not part of the definition of that element. For example, a hanging node on a triangle would be a node that lies somewhere along one of the edges of the triangle but is not one of the three vertices that defines the triangle. An example of a *non-conforming mesh* (i.e., it contains hanging nodes) is shown in Fig. 3.18. Most of the basis functions used in standard CEM analysis are not designed to work for non-conforming meshes. As a result, it is important when using a mesh generation tool to ensure that the way you are using it generates conforming meshes. An example of a CEM method that can operate on non-conforming meshes is the discontinuous Galerkin time domain (DGTD) method. One benefit of methods that operate on non-conforming meshes is that the mesh generation is much simpler, especially in the case of adaptive mesh refinement (we will discuss this more later). However, these methods often are more complicated to code and can have other advantages and disadvantages, so they are still a matter of active research interest.

Another way to classify a mesh is whether it is *structured* or *unstructured*. A structured mesh would include a regular grid of nodes that are equally spaced throughout the entire region of interest (like what is done for a basic finite difference method). An unstructured grid simply means that the nodes are placed relatively arbitrarily throughout the region of interest based off of the meshing algorithm being used to mesh the region. An unstructured grid is almost always used for FEM analysis.

Typically, deciding whether a mesh will produce "good" results for a particular application can be challenging. General "rules of thumb" exist within the CEM community that are applicable to many use cases. One common rule of thumb is that if the basis functions being used provide a linear interpolation accuracy then a mesh should typically contain mesh elements that are smaller than $\lambda/10$, where λ is the wavelength at the highest frequency of interest to be modeled. Although this rule of thumb is typically sufficient, certain features in a geometry may cause regions where the electric or magnetic fields change very rapidly as a function of position (e.g., near sharp conducting edges, between closely spaced conductors,

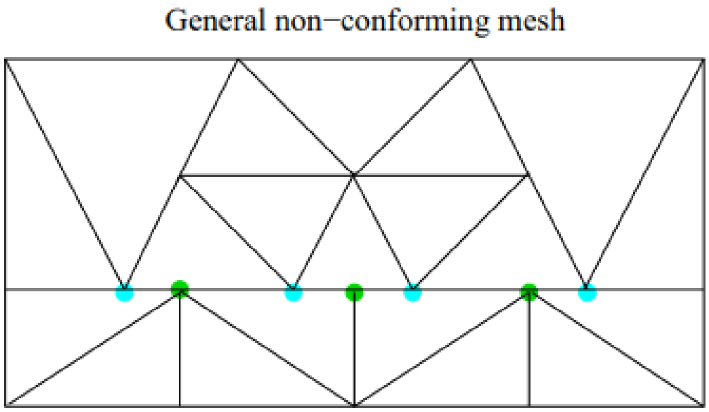


Figure 3.18: Example of a non-conforming mesh with a few of the hanging nodes shown in color (image from [26]).

etc.). Near these features a finer mesh is often needed to achieve accurate results.

Generally, performing a refinement of a mesh by hand is impractical due to its tedious and time-consuming nature. One of the most valuable features of many commercially available CEM tools is that they can perform *adaptive mesh refinement* with little interaction from the user. These adaptive algorithms start with a very coarse mesh, solve the problem, refine the mesh (based on some kind of built-in criterion of the adaptive algorithm), and then solve the problem again. The solutions for the different meshes can be compared in various ways to make decisions on how to continue to refine the mesh until a satisfactory level of convergence is achieved. This adaptive process typically leads to very accurate results, and also balances efficiency of the overall algorithm well because the adaptivity is aimed at only increasing the number of mesh elements at locations where they are needed, rather than arbitrarily refining the mesh throughout the entire region of interest.

The final detail of meshing that we will discuss is related to the *quality* of the elements. Typically, we want mesh elements to have a fairly uniform aspect ratio for the different sides and angles of the element. A comparison of good and bad quality elements is shown in Fig. 3.19. Generally, we want good quality elements because this leads to the interpolation functions used in our basis functions and numerical integration routines often used in numerical analysis to perform better over the range of the element. This effect is typically amplified for higher-order elements that use higher polynomial orders for their interpolation functions.

3.13.2 Mesh Generation Tools

For your FEM course project you will need to generate a mesh. We suggest here a few tools that can be useful in this process. If you are writing your code in Matlab, you can install the PDE toolbox and use the `generateMesh()` function to produce your mesh. The data provided by this tool can also typically give you information about which nodes or edges lie on the boundary of the region that was meshed, which is very helpful for enforcing boundary conditions correctly in the FEM process. You can also look into functions that perform a Delaunay triangulation (in 2D or 3D) to generate a mesh.

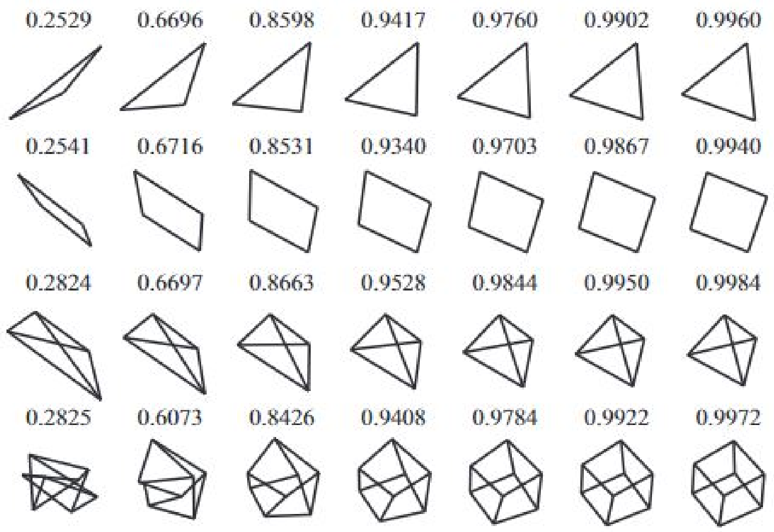


Figure 3.19: Examples of elements with different levels of quality. Low quality elements are to the left of the figure, and the mesh quality progressively gets better as the figure moves to the right (image from [27]).

If you are willing to download and install a complete meshing program, Coreform Cubit has a free student version that will be able to easily produce the kinds of meshes needed for your course project. This tool is fairly intuitive for generating basic geometries and then meshing them. It can also be configured in such a way so that information about the nodes, edges, or triangles that lie on any boundary curve or surface can be separately "tagged" to make enforcing boundary conditions easier in the FEM process. When using this tool, it is recommended that you export your mesh using the Abaqus file format. This file type stores the data in an ASCII format and so is easy to then be read into a Matlab or Python.

Another popular meshing tool is Gmsh. This is an open source tool that is freely available and is still actively being supported and developed.

3.14 Higher-order Elements

When we learned about finite difference methods, one of the first details that we considered was the *accuracy* of the different approximations of differential operators. We saw that if we were careful with which approximations we used, we could increase the accuracy from first- to second-order quite simply. Thus far, we have not really questioned how to improve the accuracy of our FEM solutions, which we will now address.

There are two basic ways to improve the accuracy of FEM solutions: using smaller element sizes (so that the underlying approximation of the field variation being linear is more appropriate) or using a higher-order polynomial interpolation function as the basis. In the context of adaptive FEM algorithms, we can choose to try and improve the solution by either refining the size of the mesh elements (known as *h-refinement*), increasing the polynomial order of the interpolation functions in an element (known as *p-refinement*), or some combination of both (usually referred to as *hp-refinement*). One of the main benefits

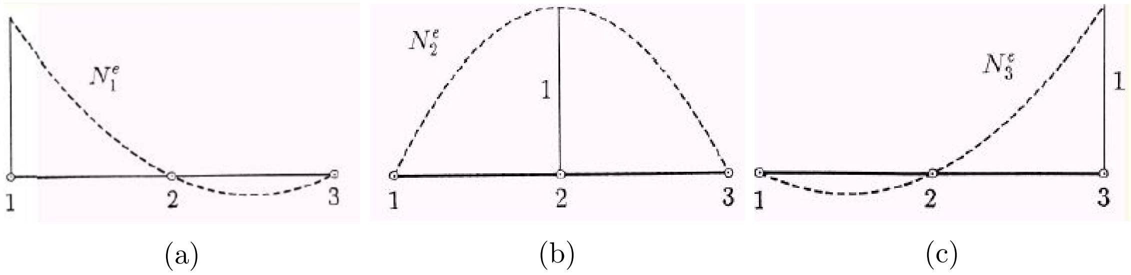


Figure 3.20: Quadratic interpolation functions for 1D finite element analysis (images from [20]).

of using higher-order elements is that because the interpolation accuracy is higher, larger element sizes can be used in the meshing process, generally leading to a smaller matrix that needs to be solved.

We will not go into detail on how to derive the higher-order interpolating functions in class, but if you are interested you can find this information in standard references like [20]. One consequence of using higher-order functions is that over each element we typically need to introduce additional nodes to interpolate at. For example, in 1D where our elements are line segments, for a second-order element each element will have three nodes (the two endpoints and typically the midpoint of the segment). Over the single element, there are now three different interpolating functions with quadratic order, as shown in Fig. 3.20. One of the consequences of these definitions are that the different basis functions used in the overall FEM formulation overlap with more nearby basis functions (their support is effectively larger). As a result, the overall sparsity of the FEM matrix reduces slightly for each increase in element order. However, this decrease in sparsity is typically well worth the improvement in overall accuracy achieved using higher-order elements. As a result, allowing some amount of support for higher-order FEM elements is very popular (e.g., many commercial FEM tools support this).

To see the value of using higher-order elements, it is customary to look at the phase error that occurs in wave propagation per wavelength for the different basis functions. This is usually done because the phase error is easy to calculate, and can also be particularly problematic in applications as it will accumulate over the span of a simulation region and can potentially lead to large errors if not appropriately controlled. The per wavelength phase error is shown in Fig. 3.21 for a 1D FEM analysis. It is clearly seen that increasing the element order can significantly increase the convergence rate compared to h-refinement (i.e., making the elements smaller). An analysis of the numerical dispersion provides the important takeaway that the convergence rate versus element order scales as $(h/\lambda)^{2p}$, where h is the average mesh element length and p is the polynomial order.

These results and trends also extend to FEM analysis in higher dimensions (i.e., 2D and 3D). In these situations, the number of nodes that are used in an element continue to increase with increasing element order. This can also result in interpolation nodes being placed at interior points of the element (i.e., they do not lie on any of the edges of the basic mesh).

One detail that does change in higher dimensions is that the structure/direction of the mesh can also impact the phase error that occurs based on the propagation direction of

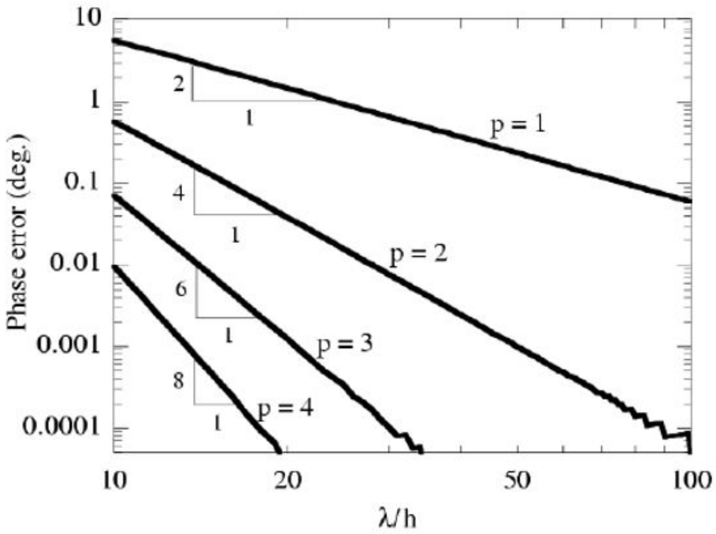


Figure 3.21: Comparison of convergence rates for 1D FEM using different orders of polynomial interpolation (denoted by p) (image from [20]).

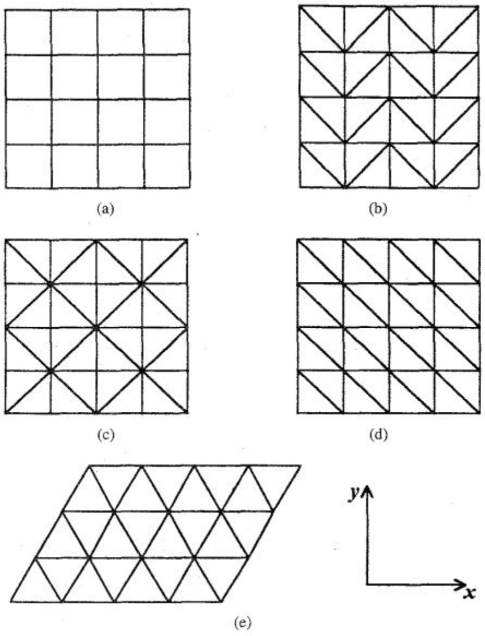


Figure 3.22: Different mesh types used in analyzing the dispersion error. (a) quadrilateral, (b) arrow, (c) diamond, (d) one-directional, and (e) hexagonal meshes (images from [28]).

the wave. We have already seen this happen to some degree in our analysis of numerical dispersion with finite difference methods. In general, having less of a regular structure to the mesh (i.e., not having all the elements pointing in a particular direction) leads to better results because the different errors that occur can sometimes cancel rather than consistently accumulating. Some basic results on this effect, as well as the benefit of using higher-order elements in higher dimensions are summarized in Figs. 3.22 to 3.24.

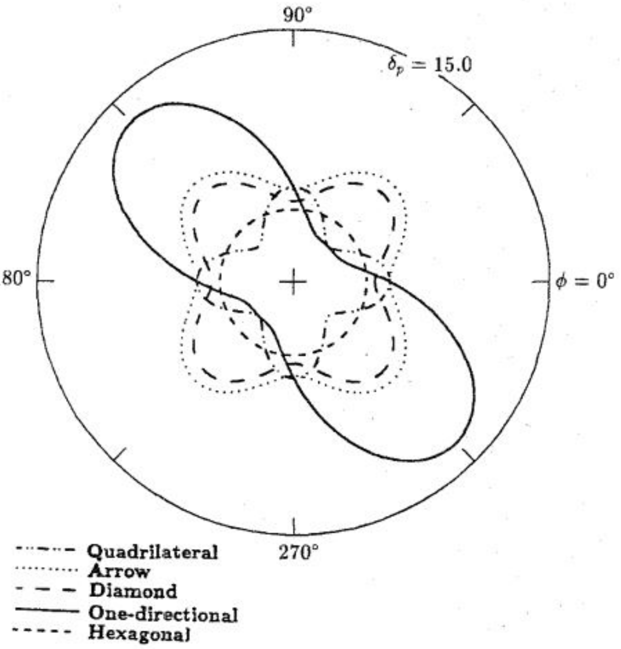


Figure 3.23: Phase error vs. propagation direction for the different meshes shown in Fig. 3.22 (image from [28]).

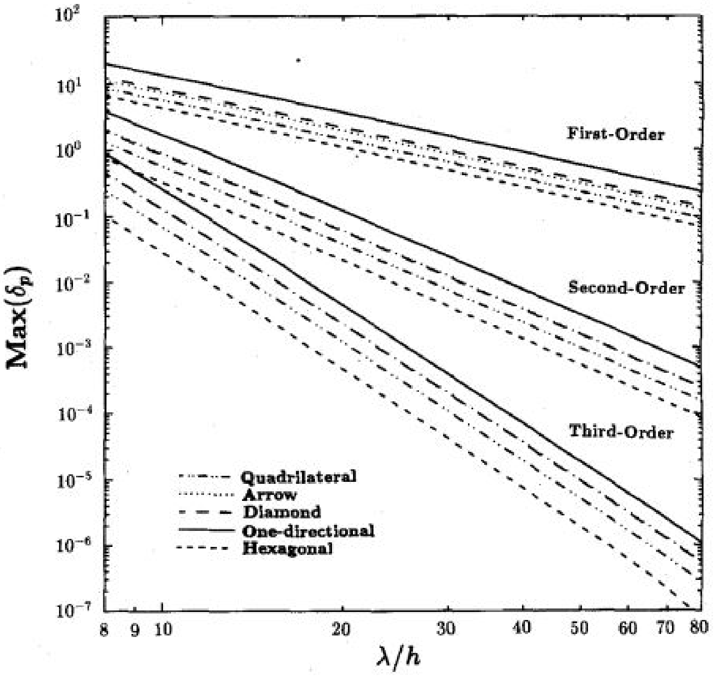


Figure 3.24: Maximum phase error vs. element order for the different meshes shown in Fig. 3.22 (image from [28]).

So far, we have primarily discussed the benefits of higher-order elements. Although higher-order elements can typically be very beneficial, there are some drawbacks. For instance, near a field singularity or other regions where the fields change very rapidly a higher polynomial order in the approximation can actually do a poor job in representing the underlying field variation [5]. In situations like this, it can be better to use smaller elements with linear interpolating functions rather than trying to increase the polynomial order. This is part of the reason why adaptive *hp*-refinement methods can be of significant interest for achieving the best performance possible.

3.15 Finite Element Method Project

This project covers the implementation of a computer code using the finite element method to solve problems in electromagnetics. A list of suggested project topics are included later in this document. The main deliverable for this project will be a written formal report that details the work that was completed. At a high-level, this report will cover the formulation of the mathematical problem solved, the discretization approach used, and a discussion of the validation of the computer code via numerical results generated. A detailed grading rubric for this report is included later in this document.

3.15.1 Suggested Project Topics

Note: If you did one of these projects for the finite difference method project, you must do a different project for your finite element method project.

1. Develop a 2D FEM program to calculate the radiation of an infinitely long electric current in an open region that contains different inhomogeneities. The open region should be terminated using an ABC. After validating that the source radiates correctly in a homogeneous open region, use your code to study **at least two** of the following:

   (a) The diffraction pattern produced by an infinitely long current source radiating in the presence of an infinitely long conducting sheet with one slot. Compare the diffraction pattern of this case with that of an infinitely long conducting sheet with two or more slots.

   (b) The scattering produced when an infinitely long current source radiates in the presence of an infinitely long conducting cylinder of various cross sections (e.g., rectangular, circular, etc.).

   (c) The scattering produced when an infinitely long current source radiates in the presence of an infinitely long dielectric cylinder of various cross sections (e.g., rectangular, circular, etc.) and material properties.

   (d) Compare the performance of **at least 2** approximate boundary conditions to terminate the open region. Examples would include first-order ABCs for a rectangular surface or an ABC that accounts for the curvature of a cylindrical surface. You may also consider higher-order versions of these ABCs. You may also consider the implementation of a PML to handle this. **Completing this item can yield up to 5 points of extra credit to the total project score.**

2. Solve Laplace's equation for various "shielded" transmission line structures that support TEM or quasi-TEM modes. Validate that your code is working by considering at least one geometry where reasonable analytical formulas exist for the line capacitance (e.g., a coaxial line or a stripline). For the geometries studied, plot the equipotential lines and static electric field distribution. For transmission lines that are not naturally "shielded" (e.g., a microstrip trace or a coplanar waveguide), ensure that the "extra" shield conductors are placed far enough away from the desired parts of the transmission line geometry that they minimally affect the solution. Possible transmission lines to study include: coaxial line, microstrip line, stripline, coplanar waveguide, grounded coplanar waveguide, slotline, etc.

3. Develop a 2D FEM program to calculate the field modal distributions and propagation constants of empty rectangular and circular waveguides. Plot the mode distributions and dispersion curves for the first three TE and TM modes (i.e., 6 modes total). For the rectangular waveguide, use an approximately 2:1 ratio for the lengths of the rectangle. After validating your results, simulate a more complex empty waveguide, such as different kinds of ridged waveguides.

4. Develop an FEM program to calculate the dispersion curve and transverse electric field distribution of a partially-filled rectangular waveguide. Validate the results of the dispersion curve against the analytical solution (see Sec. 5.3.2 of your textbook for the analytical solution). **Completing this item can yield up to 10 points of extra credit to the total project score.**

5. Use the finite element method to solve one problem of interest to you. Make sure to plan for some way to validate your code's performance for your selected problem.

3.15.2 Rubric

1. Title & Abstract (5 points)

   (a) Title and abstract are concise, but informative.

   (b) Abstract should properly convey the main information contained in the work, the methods used, and the problems studied.

2. Introduction and Conclusion (10 points)

   (a) Introduction should discuss relevant background and history of the problem to be studied and the methods used in the work, supported by relevant references from textbooks and the literature (around 4 or 5 references is likely plenty for this report). Introduction should also finish with a paragraph discussing the organization of the remainder of the paper.

   (b) Conclusion should succinctly summarize the content of the work and mention possible directions for further study, improvements that could be made to the numerical methods, etc.

3. Formulation & Discretization (30 points)

   (a) Equations that are to be solved numerically are appropriately derived from a well-established starting point (e.g., Maxwell's equations).
   (b) Assumptions or approximations of the derivation are clearly communicated.
   (c) Basic process of the numerical discretization is clearly communicated for all important/distinct equations.

4. Numerical Results (45 points)
   (a) Validation data is shown to demonstrate correct implementation of the numerical method. Sufficient details on the numerical results and validation data should also be included so that someone else could conceivably implement their own tool and replicate your results. Sample items to cover would be sizes of the simulation region and any objects involved, average element size, relative permittivity and permeability of materials, etc. (Note: this is not an exhaustive list of what should be covered).
   (b) Additional numerical results are presented to show utility of the numerical method. Again, sufficient detail is provided for simulation parameters that a reader can understand the content of the simulation and recreate it themselves.
   (c) Figures are legible and aesthetically-pleasing (Matlab/Python plots are fine). Figure captions are concise, but informative. Figures are referenced and discussed appropriately within the text of the report.
   (d) Note: your code must correctly implement the numerical method to approach reaching full points in this category of the rubric.

5. Writing Style (5 points)
   (a) Grammar, word use, spelling, etc. are of an overall good quality.
   (b) Best practices for writing mathematical prose are followed (equations are treated as part of the sentence, equations are numbered, "user-friendly" references to previous equations, etc.). See "What's Wrong with these Equations?" by N. David Mermin for basic guidelines to consider.
   (c) Equations are typeset in an aesthetically-pleasing manner.
   (d) Note: if the writing style is particularly poor, additional points will be subtracted from other aspects of the report (e.g., Formulation & Discretization or Numerical Results).

6. Coding Style (5 points)
   (a) Code is formatted and organized in an easily-readable manner. Descriptive variable and function names are used as appropriate.
   (b) Sufficient comments are used to make the code more easily interpreted by another person.

Chapter 4

Method of Moments and Fast Algorithms

4.1 Introduction to the Method of Moments

The final main CEM technique we will learn about in this course is the *method of moments (MoM)*, which is also sometimes referred to as the *moment method* or the *boundary element method (BEM)*. At a high-level, MoM follows essentially the same process as FEM but is applied to integral equations rather than differential equations. Although this seems like a "small change", the formulation and solution of integral equations is so different from working with differential equations to the point that the MoM is well and truly a distinct computational technique to study.

To gain some insight into the MoM, we will begin by studying how it can be applied to the electrostatics problem of computing the capacitance of a PEC structure embedded in a homogeneous medium. We will need to begin by formulating an appropriate integral equation. Generally, there are a number of different ways to derive an integral equation for a particular problem that will yield equivalent results. We will not be rigorous in our approach here for the electrostatic problem to more quickly illustrate how the MoM works, but will be more rigorous later when we consider solving the wave equation.

4.1.1 Green's functions

Electromagnetic integral equations are generally cast in the form of some kind of surface source being integrated against a Green's function for a related problem. Considering this, we will first take some time to review how Green's functions arise in basic electrostatic theory. For our electrostatics problem of interest, the differential equation that will be solved is Poisson's equation, which is

$$\nabla^2\Phi(\mathbf{r}) = -\rho/\epsilon, \tag{4.1}$$

where ρ is the charge density and ϵ is the background permittivity of a homogeneous region the charge exists in. We can also define a Green's function for this problem that satisfies

(4.1) for a *point source excitation.* This is

$$\nabla^2 g(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}'), \tag{4.2}$$

where $g(\mathbf{r}, \mathbf{r}')$ is the Green's function and $\delta(\mathbf{r} - \mathbf{r}')$ is the Dirac delta function. Using standard methods, it is possible to determine that the solution to (4.2) is

$$g(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|}. \tag{4.3}$$

Even if you are unfamiliar with discussing this problem in terms of Green's functions, you have encountered them before in your study of electromagnetics. As an example of this, we can consider Coulomb's law, which told us that the electric field produced by a point charge was

$$\mathbf{E}(\mathbf{r}) = \hat{r}\frac{q}{4\pi\epsilon|\mathbf{r}|^2}. \tag{4.4}$$

Recognizing that $\mathbf{E} = -\nabla\Phi$, we can conclude that the potential would be

$$\Phi(\mathbf{r}) = \frac{q}{4\pi\epsilon|\mathbf{r}|}. \tag{4.5}$$

To be more general in our description, it is best that we allow our point charge to be located at a position other than 0. This generalization is possible by writing the potential as

$$\Phi(\mathbf{r}) = \frac{q(\mathbf{r}')}{4\pi\epsilon|\mathbf{r} - \mathbf{r}'|}, \tag{4.6}$$

where $\mathbf{r}$ and $\mathbf{r}'$ are position vectors pointing to the observation and source points, respectively.

If instead of having a single point charge we had a distribution, you will recall that our formula for the potential generalizes to

$$\Phi(\mathbf{r}) = \iiint \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|}\rho(\mathbf{r}')/\epsilon\, dV', \tag{4.7}$$

where the prime on dV' denotes that we are integrating with respect to the primed variables. We can interpret this formula as being a superposition of the potentials produced by a collection of point charges. A common terminology is to refer to the

$$\frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} \tag{4.8}$$

portion of this integral as an *integration kernel.* The reason for this is because this is the *point source response* that we are going to integrate against our distribution to determine the potential at any other location. We also see that this point source response exactly matches our Green's function given earlier.

Although Green's functions are useful as a "point source response", it is also valuable to us because we can view it as giving us a way to "invert" differential equations. To see this, we can integrate Poisson's equation against the Green's function to get

$$\iiint_V g(\mathbf{r}, \mathbf{r}')\nabla'^2\Phi(\mathbf{r}')\, dV' = -\iiint_V g(\mathbf{r}, \mathbf{r}')\rho(\mathbf{r}')/\epsilon\, dV'. \tag{4.9}$$

We can now go about transferring the derivatives from Φ onto $g(\mathbf{r}, \mathbf{r}')$ in (4.9) using integration by parts. In a homogeneous region, this gives us

$$\begin{aligned}\iiint_V g(\mathbf{r}, \mathbf{r}')\nabla'^2\Phi(\mathbf{r}')\,dV' &= -\iiint_V \nabla' g(\mathbf{r}, \mathbf{r}')\cdot\nabla'\Phi(\mathbf{r}')\,dV' \\ &= \iiint_V \left[\nabla'^2 g(\mathbf{r}, \mathbf{r}')\right]\Phi(\mathbf{r}')\,dV',\end{aligned} \tag{4.10}$$

where we have ignored the boundary terms from the integration by parts because they are located at infinity where these functions go to 0. We can now make use of the fact that

$$\nabla' g(\mathbf{r}, \mathbf{r}') = -\nabla g(\mathbf{r}, \mathbf{r}') \tag{4.11}$$

for our Green's function given in (4.3) twice to write the last line of (4.10) as

$$\iiint_V \left[\nabla'^2 g(\mathbf{r}, \mathbf{r}')\right]\Phi(\mathbf{r}')\,dV' = \iiint_V \left[\nabla^2 g(\mathbf{r}, \mathbf{r}')\right]\Phi(\mathbf{r}')\,dV'. \tag{4.12}$$

Our final step is to use the definition of the Green's function via the differential equation it satisfies given in (4.2) to note that

$$\iiint_V \left[\nabla^2 g(\mathbf{r}, \mathbf{r}')\right]\Phi(\mathbf{r}')\,dV' = -\iiint_V \delta(\mathbf{r}-\mathbf{r}')\Phi(\mathbf{r}')\,dV' = -\Phi(\mathbf{r}). \tag{4.13}$$

We may set this equal to the right-hand side of (4.9) to recover Coulomb's law. From this, we see that Coulomb's law is the "inverse" of Poisson's equation facilitated via the Green's function.

4.1.2 Electrostatic Integral Equation

Now that we have some more familiarity of working with the Green's function, we can consider the integral equation that we will want to solve via the MoM. We begin by assuming that we have a perfect conductor that has some unknown surface charge distribution. The potential produced at an observation point $\mathbf{r}$ by this surface charge distribution can be determined from Coulomb's law to be

$$\iint_S \epsilon^{-1} g(\mathbf{r}, \mathbf{r}')\rho_s(\mathbf{r}')dS' = \Phi(\mathbf{r}), \tag{4.14}$$

where S is the surface of the metallic object and dS' means we are integrating over the primed variables on surface S in this equation. Since $\Phi(\mathbf{r})$ is not known at general $\mathbf{r}$ and ρ_s is also not known over S, we seem to be at somewhat of an impasse on how to proceed. The solution is to take our observation point $\mathbf{r}$ to be on the surface of the conductor where we know that the potential must be a known constant value (this known value denoted as Φ_0 comes from the boundary condition of the problem). In this case, our integral equation becomes

$$\iint_S \epsilon^{-1} g(\mathbf{r}, \mathbf{r}')\rho_s(\mathbf{r}')dS' = \Phi_0, \quad \mathbf{r} \in S. \tag{4.15}$$

We can now go about solving this integral equation for ρ_s by following the same basic process as the FEM. Namely, we can subdivide the surface S up into smaller portions where we can express ρ_s using simple basis functions. We then get that

$$\rho_s(\mathbf{r}') = \sum_{n=1}^{N} c_n v_n(\mathbf{r}'), \tag{4.16}$$

where v_n is the basis function and c_n is the corresponding expansion coefficient. If we substitute this into (4.15), we arrive at

$$\sum_{n=1}^{N} c_n \iint_S \epsilon^{-1} g(\mathbf{r}, \mathbf{r}') v_n(\mathbf{r}') dS' = \Phi_0, \quad \mathbf{r} \in S. \tag{4.17}$$

We are now faced with the same problem that we had in our FEM formulation; namely, we have a finite number of degrees of freedom to solve for but our integral equation is still in a continuous, infinite-dimensional form since it must be satisfied at every $\mathbf{r}$ on the surface S. This can be solved in the same way as with FEM by *testing* the equation using some set of weighting or testing functions. If we denote our set of testing functions as $w_m(\mathbf{r})$ with $m \in [1, N]$ and test (4.17) by integrating over S after multiplying by w_m, we get

$$\sum_{n=1}^{N} c_n \iint_S w_m(\mathbf{r}) \iint_S \epsilon^{-1} g(\mathbf{r}, \mathbf{r}') v_n(\mathbf{r}') dS' dS = \iint_S w_m(\mathbf{r}) \Phi_0 dS. \tag{4.18}$$

We may do this for all m and assemble the resulting set of equations into a matrix of the form

$$[A]\{c\} = \{b\}, \tag{4.19}$$

where

$$[A]_{mn} = \iint_S \iint_S w_m(\mathbf{r}) \epsilon^{-1} g(\mathbf{r}, \mathbf{r}') v_n(\mathbf{r}') dS' dS, \tag{4.20}$$

$$\{b\}_m = \iint_S w_m(\mathbf{r}) \Phi_0 dS. \tag{4.21}$$

We can solve this matrix equation to recover the surface charge density. With the surface charge density, we can then numerically evaluate Coulomb's law to determine the potential at any point in space. We can also compute the total surface charge and divide by the applied potential Φ_0 to compute the capacitance as $C = Q/\Phi_0$, where Q is the total surface charge.

Of course, this all depends on choosing suitable functions for v_n and w_m. In our FEM analysis there were only a relatively "limited" number of functions that we could use. In contrast to this, the space of functions that suitable MoM discretizations can be achieved with is much larger. For instance, because we do not have to evaluate any derivatives of our basis or testing functions in (4.20), we can use basis or testing functions with lower order than the first-order functions that were suitable for the FEM. As an example, we can choose to use zeroth-order basis functions that are constant on a particular patch of the mesh and zero elsewhere; e.g.,

$$v_n(\mathbf{r}') = \begin{cases} 1, & \mathbf{r}' \in S_n \\ 0, & \text{elsewhere}, \end{cases} \tag{4.22}$$

where S_n is the nth surface patch (or cell) of the mesh. This function is sometimes referred to as a *pulse basis* in the CEM literature.

The choice of testing functions is likewise enlarged compared to FEM. The simplest option that we can select is to test our equation with a Dirac delta function. For instance, a typical approach for this strategy would be to use

$$w_m(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_m), \tag{4.23}$$

where $\mathbf{r}_m$ is the center of the mth cell of the mesh. This approach is usually referred to as *point collocation* or *point matching* since it is equivalent to enforcing our integral equation only at the center of each cell of the mesh. For this case, (4.20) becomes

$$[A]_{mn} = \iint_{S_n} \epsilon^{-1} g(\mathbf{r}_m, \mathbf{r}') dS' \tag{4.24}$$

and $\{b\}_m = \Phi_0$.

If $m \neq n$, we can use simple integration methods to evaluate (4.24); e.g., the midpoint integration rule. When $m = n$, we are faced with the difficulty that we need to integrate over the Green's functions' singularity of $1/|\mathbf{r}_m - \mathbf{r}'|$ where $\mathbf{r}'$ can equal $\mathbf{r}_m$. For this simple integral equation, it can be possible to use somewhat simple analytical techniques to evaluate this singular integral. However, this is not typically the case for the full 3D case of solving dynamic electromagnetic problems where much more sophisticated methods are needed. For this simple electrostatic case, we can approximate S_n as a circular disc with the same area as S_n and evaluate the potential at its center. Using these various approximations, we arrive at

$$[A]_{mn} = \begin{cases} \dfrac{S_n}{4\pi\epsilon|\mathbf{r}_m - \mathbf{r}_n|}, & m \neq n \\ \\ \dfrac{1}{2\epsilon}\sqrt{\dfrac{S_n}{\pi}}, & m = n. \end{cases} \tag{4.25}$$

As an alternative approach, we may use a pulse function as testing instead of a delta function. In this situation,

$$[A]_{mn} = \iint_{S_m} \iint_{S_n} \epsilon^{-1} g(\mathbf{r}, \mathbf{r}') dS' dS \tag{4.26}$$

and $\{b\}_m = \Phi_0 S_m$ where S_m is the area of the mth surface patch of the mesh. One benefit of this approach is that the system matrix will now be symmetric, which opens the door to using specialized numerical linear algebra techniques in solving (4.26). However, we will now need to deal with the singularity in our integral equation more carefully due to the double surface integration involved when $m = n$. We will briefly discuss how to deal with this kind of case for more realistic problems later in the course.

4.1.3 FEM and MoM Differences

As mentioned previously, we can see that the basic discretization process of the MoM exactly matches the *weighted residual method* that we discussed in the context of FEM. Although these similarities exist, there are many important differences between FEM and MoM due to the differences in solving differential and integral equations. Some of the main differences are the following.

1. When solving differential equations, we had to discretize the entire volume of interest. Integral equations provide us with an approach that only requires discretizing the surface of the object of interest. For particularly large problems, this reduction in dimensionality can be quite important.

2. Related to 1., when we solved differential equations we had to determine artificial boundary conditions to terminate open region problems (e.g., ABCs or PMLs). These approximate boundary conditions contribute to numerical error. In contrast to this, our integral equation does not require any kind of artificial boundary to be imposed because the Green's function used in our problem formulation automatically encodes the correct behavior into our solutions.

3. For FEM, the system matrix was extremely sparse. For MoM, the system matrix is completely *dense*, i.e., every element in the matrix is nominally non-zero. The reason for this is that the Green's function as the integration kernel is able to link every single basis and testing function to one another since it is not a purely "local" operator. The fact that MoM produces dense matrices greatly changes the numerical linear algebra solution approaches that should be used when solving MoM matrix equations versus FEM matrix equations.

4. We often found that the PDEs we wished to solve with the FEM were self-adjoint problems so that the Galerkin method typically led to a very well-performing discretization. This will *not* always be the case for integral equations.

5. Related to 4., the types of basis and testing functions that can be used in the MoM discretization constitutes a much larger set than is admissible for FEM. Even though this is the case, the modern CEM literature has mostly agreed upon the most useful basis and testing functions to be used for most 3D electromagnetic problems so that this full flexibility is not necessarily exploited.

6. For FEM, we could sometimes evaluate the integrals needed in evaluating matrix elements analytically or at worst using fairly simple numerical integration routines. The presence of the $1/|\mathbf{r}-\mathbf{r}'|$ singularity in the Green's function for electromagnetic integral equations makes both the analytical and numerical evaluation of these integrals much harder.

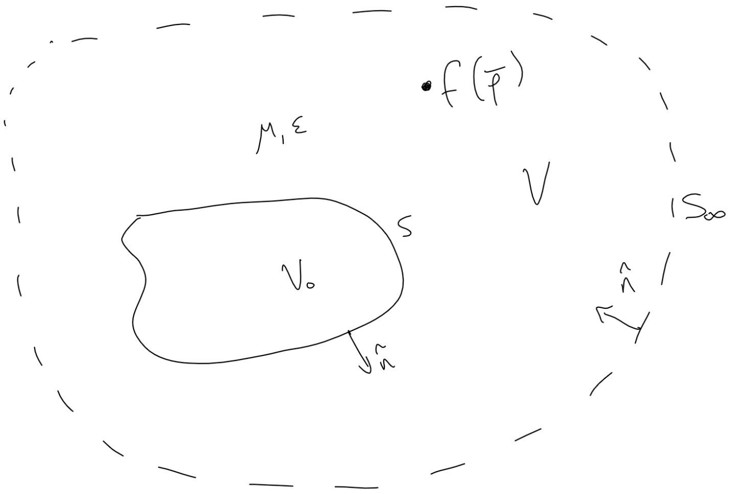


Figure 4.1: Illustration of the problem scenario to derive an integral equation.

4.2 Formulation of Integral Equation for 2D Helmholtz Equation

We will now consider the more formal derivation of an integral equation that can be used to solve problems related to the 2D Helmholtz equation. This will initially follow a very general procedure that will lead to a result that we can apply more specific conditions to later in order to derive integral equations for particular problems of interest.

To begin, we will assume that we have a scalar wave $\varphi(\boldsymbol{\rho})$ that is produced by a source $f(\boldsymbol{\rho})$ in the presence of an arbitrarily shaped object with exterior surface S immersed in a homogeneous background material characterized by μ and ϵ. For our initial derivation, we will not need to consider the properties of the material (e.g., dielectric or conductive properties) that is contained within S, but we will return to this point when we formulate integral equations applicable to more specific scenarios. This setup is illustrated in Fig. 4.1.

For the problem illustrated in Fig. 4.1, our wave function $\varphi(\boldsymbol{\rho})$ will satisfy the inhomogeneous Helmholtz equation exterior to S; i.e.,

$$\nabla^2\varphi(\boldsymbol{\rho}) + k^2\varphi(\boldsymbol{\rho}) = f(\boldsymbol{\rho}), \quad \boldsymbol{\rho} \in V. \tag{4.27}$$

Additionally, for this kind of open region problem the wave function will also satisfy the *radiation condition*

$$\sqrt{\rho}\big[\partial_\rho\varphi(\boldsymbol{\rho}) + jk\varphi(\boldsymbol{\rho})\big] = 0, \quad \rho \to \infty. \tag{4.28}$$

This condition indicates that the wave propagates toward "infinity" without reflection and that the field values decay to 0 as they reach "infinity". As alluded to previously, we will also need to make use of the Green's function for this problem. The Green's function will satisfy

$$\nabla^2 g(\boldsymbol{\rho}, \boldsymbol{\rho}') + k^2 g(\boldsymbol{\rho}, \boldsymbol{\rho}') = \delta(\boldsymbol{\rho} - \boldsymbol{\rho}'), \tag{4.29}$$

as well as the radiation condition given in (4.28). The solution to (4.29) can be written in terms of the well-known Hankel functions. In particular, we have

$$g(\boldsymbol{\rho}, \boldsymbol{\rho}') = \frac{1}{4j} H_0^{(2)}(k|\boldsymbol{\rho} - \boldsymbol{\rho}'|) \tag{4.30}$$

where $H_0^{(2)}$ is the zeroth-order Hankel function of the second kind (which describes an outgoing cylindrical wave).

To derive our integral equation, we now multiply (4.27) by $g(\boldsymbol{\rho}, \boldsymbol{\rho}')$ and (4.29) by $\varphi(\boldsymbol{\rho})$ and integrate the difference of the two equations over the entire exterior region of V illustrated in Fig. 4.1. This yields

$$\int_V \left[g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla^2\varphi(\boldsymbol{\rho}) - \varphi(\boldsymbol{\rho})\nabla^2 g(\boldsymbol{\rho}, \boldsymbol{\rho}') \right] dV = \int_V g(\boldsymbol{\rho}, \boldsymbol{\rho}') f(\boldsymbol{\rho}) dV + \int_V \varphi(\boldsymbol{\rho})\delta(\boldsymbol{\rho} - \boldsymbol{\rho}')dV. \tag{4.31}$$

Before continuing, we will quickly simplify the right-hand side of (4.31). We first note that the first integral is the homogeneous medium Green's function integrated against the source distribution, so this will simply produce a field due to the source distribution $f(\boldsymbol{\rho})$ as if the inhomogeneous scattering object specified by S was not present. We typically refer to this basic concept as the *incident field*, with the explicit mathematical relationship being

$$\varphi_{\text{inc}}(\boldsymbol{\rho}') = -\int_V g(\boldsymbol{\rho}, \boldsymbol{\rho}') f(\boldsymbol{\rho}) dV. \tag{4.32}$$

Using this result, we have that (4.31) becomes

$$\int_V \left[g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla^2\varphi(\boldsymbol{\rho}) - \varphi(\boldsymbol{\rho})\nabla^2 g(\boldsymbol{\rho}, \boldsymbol{\rho}') \right] dV = -\varphi_{\text{inc}}(\boldsymbol{\rho}') + \int_V \varphi(\boldsymbol{\rho})\delta(\boldsymbol{\rho} - \boldsymbol{\rho}')dV. \tag{4.33}$$

We may now continue our derivation by noting that

$$\nabla \cdot \Big(g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla\varphi(\boldsymbol{\rho}) - \varphi(\boldsymbol{\rho})\nabla g(\boldsymbol{\rho}, \boldsymbol{\rho}') \Big) = g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla^2\varphi(\boldsymbol{\rho}) - \varphi(\boldsymbol{\rho})\nabla^2 g(\boldsymbol{\rho}, \boldsymbol{\rho}'), \tag{4.34}$$

so that we can use Gauss' theorem on the left-hand side of (4.33) to get

$$\begin{aligned} -\oint_S \hat{n} \cdot \left[g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla\varphi(\boldsymbol{\rho}) - \varphi(\boldsymbol{\rho})\nabla g(\boldsymbol{\rho}, \boldsymbol{\rho}') \right] dS - \oint_{S_\infty} \hat{n} \cdot \left[g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla\varphi(\boldsymbol{\rho}) - \varphi(\boldsymbol{\rho})\nabla g(\boldsymbol{\rho}, \boldsymbol{\rho}') \right] dS \\ = -\varphi_{\text{inc}}(\boldsymbol{\rho}') + \int_V \varphi(\boldsymbol{\rho})\delta(\boldsymbol{\rho} - \boldsymbol{\rho}')dV. \end{aligned} \tag{4.35}$$

Note that the minus signs on the left-hand side are due to the direction of our unit normal vectors in Fig. 4.1 being in the opposite direction to those assumed in Gauss' theorem. Now, since both $\varphi(\boldsymbol{\rho})$ and $g(\boldsymbol{\rho}, \boldsymbol{\rho}')$ satisfy the radiation condition given in (4.28), the integral at S_∞ vanishes leaving us with

$$\oint_S \hat{n} \cdot \left[\varphi(\boldsymbol{\rho})\nabla g(\boldsymbol{\rho}, \boldsymbol{\rho}') - g(\boldsymbol{\rho}, \boldsymbol{\rho}')\nabla\varphi(\boldsymbol{\rho}) \right] dS + \varphi_{\text{inc}}(\boldsymbol{\rho}') = \int_V \varphi(\boldsymbol{\rho})\delta(\boldsymbol{\rho} - \boldsymbol{\rho}')dV. \tag{4.36}$$

We may now use the definition of the delta function to note that

$$\oint_S \hat{n} \cdot \left[\varphi(\boldsymbol{\rho})\nabla g(\boldsymbol{\rho},\boldsymbol{\rho}') - g(\boldsymbol{\rho},\boldsymbol{\rho}')\nabla\varphi(\boldsymbol{\rho})\right] dS + \varphi_{\text{inc}}(\boldsymbol{\rho}') = \begin{cases} \varphi(\boldsymbol{\rho}'), & \boldsymbol{\rho}' \in V, \\ 0, & \boldsymbol{\rho}' \in V_0. \end{cases} \tag{4.37}$$

This result forms the foundation of deriving integral equations for 2D electromagnetic wave problems. However, it is traditional to use the symmetry of the Green's function at this point to interchange primed and unprimed coordinates in (4.37) so that our result follows the more typical notation of primed coordinates acting as "source points" and the unprimed coordinates acting as "observation points". Doing this, we get

$$\oint_S \hat{n}' \cdot \left[\varphi(\boldsymbol{\rho}')\nabla' g(\boldsymbol{\rho},\boldsymbol{\rho}') - g(\boldsymbol{\rho},\boldsymbol{\rho}')\nabla'\varphi(\boldsymbol{\rho}')\right] dS' + \varphi_{\text{inc}}(\boldsymbol{\rho}) = \begin{cases} \varphi(\boldsymbol{\rho}), & \boldsymbol{\rho} \in V, \\ 0, & \boldsymbol{\rho} \in V_0. \end{cases} \tag{4.38}$$

Up to this point, we have predominantly just been applying generic mathematical operations that are valid but which we haven't prescribed any physical meaning to. To gain a little more insight, it will be useful to consider (4.38) a little more closely. The first point to make is that we have two unknowns in (4.38); namely, $\varphi(\boldsymbol{\rho}')$ and $\hat{n}' \cdot \nabla'\varphi(\boldsymbol{\rho}')$. We typically refer to these as *equivalent surface sources* that are induced due to the presence of the incident field. Since we currently only have a single equation derived, it should not surprise you that we will need to apply a few more conditions to arrive at a solvable integral equation. The particular conditions that should be enforced depend on the properties of the interior region of S, and we will consider a few specific examples later.

The next item to note is that once we solve for the equivalent surface sources on S, we can compute the field at any other position by evaluating (4.38). This is a mathematical representation of what is usually referred to as *Huygens' principle* for scalar fields in electromagnetics or physics. In the more mathematical literature, this kind of expression will also sometimes be referred to as an *integral representation formula* since we are expressing the solution to the differential equation using an "integral representation". A further item that we can recognize is that if we locate our observation point $\boldsymbol{\rho}$ *anywhere* within the interior volume the equivalent surface sources will produce fields that *exactly cancel* the incident field. This is sometimes referred to as the *extinction theorem* [29].

4.2.1 Bringing $\boldsymbol{\rho}$ to S

Recall from our electrostatic example, that in order to derive an integral equation from what was effectively Coulomb's law we needed to take our observation point to be on the integration surface S as part of the testing process. As mentioned at that time, doing this for general Green's functions can require some care due to the singularity involved in the Green's function. We will now take a more careful approach to dealing with this singularity in our derivation.

The main strategy is to deform our surface integral slightly to "avoid" exactly hitting the singularity and to then evaluate the resulting integral in the limit as this deformation vanishes. Different deformations can be used, but for most cases it works well in 2D to use a semicircular deformation as shown in Fig. 4.2. We can then decompose the total surface

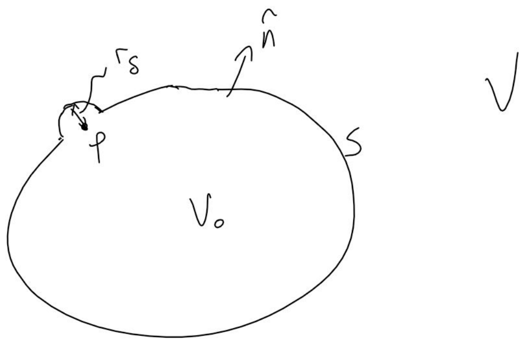


Figure 4.2: Illustration of the deformed integration path used to handle the singularity in the Green's function.

integration into the original surface minus the deformed part, denoted $S-2r_\delta$ where r_δ is the radius of the semicircular deformation, and the integration purely over the deformed part. Mathematically, we can write this decomposition as

$$\oint_S \left[\,\cdot\,\right] dS' = \lim_{r_\delta \to 0} \left\{ \int_{S-2r_\delta} \left[\,\cdot\,\right] dS' + \int_0^\pi \left[\,\cdot\,\right] r_\delta d\phi'_\delta \right\} \tag{4.39}$$

where $\left[\,\cdot\,\right]$ is a shorthand notation placeholder for the expressions in the surface integral given in (4.38). Since this deformation of an integration region is a common occurrence in mathematics and physics, it has a number of special notations for it. One is

$$\fint_S \left[\,\cdot\,\right] dS' = \lim_{r_\delta \to 0} \int_{S-2r_\delta} \left[\,\cdot\,\right] dS', \tag{4.40}$$

while another is

$$\text{P.V.} \int_S \left[\,\cdot\,\right] dS' = \lim_{r_\delta \to 0} \int_{S-2r_\delta} \left[\,\cdot\,\right] dS'. \tag{4.41}$$

In (4.41), the P.V. stands for *principal value*, since this type of integral with the deformed part taken care of separately is usually referred to as the principal value of the integration.

We will now go about evaluating the singular part of the integral. For this case, we have that

$$\lim_{r_\delta \to 0} \int_0^\pi \left[\,\cdot\,\right] r_\delta d\phi'_\delta = \frac{1}{4j} \lim_{r_\delta \to 0} \int_0^\pi \left[\varphi(\boldsymbol{\rho}') \partial_{r_\delta} H_0^{(2)}(kr_\delta) - H_0^{(2)}(kr_\delta) \partial_n \varphi(\boldsymbol{\rho}') \right] r_\delta d\phi'_\delta. \tag{4.42}$$

We can now use the small-argument approximation for the Hankel function because $r_\delta \to 0$. The small-argument approximation is

$$H_0^{(2)}(z) \approx 1 - j\frac{2}{\pi} \ln\left(\frac{\gamma z}{2}\right), \qquad z \to 0, \tag{4.43}$$

where $\gamma \approx 1.781$. Using this approximation in (4.42), yields

$$\frac{1}{4j}\lim_{r_\delta \to 0}\int_0^{\pi}\left[\varphi(\boldsymbol{\rho}')\left(-j\frac{2}{\pi}\frac{2}{\gamma k r_\delta}\frac{k\gamma}{2}\right)-\partial_n\varphi(\boldsymbol{\rho}')\left\{1-j\frac{2}{\pi}\ln\left(\frac{\gamma k r_\delta}{2}\right)\right\}\right]r_\delta d\phi'_\delta = -\frac{1}{2}\varphi(\boldsymbol{\rho}). \tag{4.44}$$

We may now use this result to consolidate our expression in (4.38) when $\boldsymbol{\rho} \in S$. However, we must first choose which version of the right-hand side to use in our final expression. The correct choice for this instance is to use the $\boldsymbol{\rho} \in V_0$ case because of the deformation we used in evaluating the singular part of the integral. Considering this, our final result for our integral equation is

$$\boxed{\int\!\!\!\!\!\!-_S \left[\varphi(\boldsymbol{\rho}')\partial_{n'}g(\boldsymbol{\rho},\boldsymbol{\rho}')-g(\boldsymbol{\rho},\boldsymbol{\rho}')\partial_{n'}\varphi(\boldsymbol{\rho}')\right]dS'+\varphi_{\text{inc}}(\boldsymbol{\rho})=\frac{1}{2}\varphi(\boldsymbol{\rho}),\quad \boldsymbol{\rho}\in S,} \tag{4.45}$$

where we have also consolidated our notation for the normal derivatives involved in the expression.

4.3 2D Electric Field Integral Equation (EFIE)

We will now look at how to use (4.45) to formulate an integral equation for a particular problem of interest. In particular, we will consider the scattering produced by a conducting cylinder of arbitrary cross section. For this problem (illustrated in Fig. 4.3), the Helmholtz equation that needs to be satisfied for the TM$_z$ polarization is

$$\nabla^2 E_z(\boldsymbol{\rho}) + k^2 E_z(\boldsymbol{\rho}) = jk\eta J_{i,z}(\boldsymbol{\rho}), \qquad \boldsymbol{\rho}\in V, \tag{4.46}$$

where $J_{i,z}$ is an impressed current source that will produce the incident field for the scattering problem. The boundary conditions that must be satisfied on the surface of the cylinder are that

$$E_z(\boldsymbol{\rho}) = 0, \qquad \boldsymbol{\rho}\in S, \tag{4.47}$$

$$\partial_n E_z(\boldsymbol{\rho}) = jk\eta H_t(\boldsymbol{\rho}) = jk\eta J_{s,z}(\boldsymbol{\rho}), \qquad \boldsymbol{\rho}\in S, \tag{4.48}$$

where H_t is the magnetic field tangential to the surface of the cylinder and $J_{s,z}$ is the surface current density induced on the cylinder due to the incident field.

We may now use these boundary conditions to simplify (4.45). In particular, if we write (4.45) specifically with the notation of this TM$_z$ problem we will have

$$\int\!\!\!\!\!\!-_S \left[E_z(\boldsymbol{\rho}')\partial_{n'}g(\boldsymbol{\rho},\boldsymbol{\rho}')-g(\boldsymbol{\rho},\boldsymbol{\rho}')\partial_{n'}E_z(\boldsymbol{\rho}')\right]dS'+E_z^{\text{inc}}(\boldsymbol{\rho})=\frac{1}{2}E_z(\boldsymbol{\rho}), \qquad \boldsymbol{\rho}\in S. \tag{4.49}$$

We can see that the Dirichlet boundary condition given in (4.47) allows us to set all E_z terms equal to 0 in (4.49). We further see that our Neumann boundary condition in (4.48) allows us to rewrite our second equivalent surface source of $\partial_n E_z(\boldsymbol{\rho})$ specifically as the actual

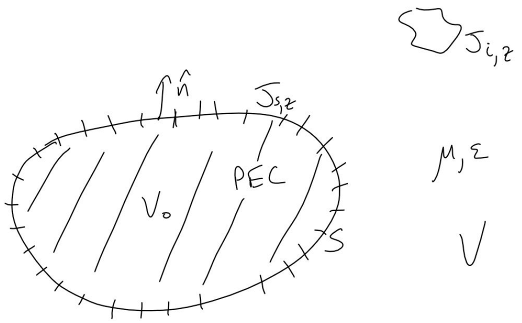


Figure 4.3: Illustration of the problem setup for TM$_z$ scattering from a conducting cylinder.

induced current density on the cylinder. From these two steps, we are able to reduce our integral equation in (4.49) that had two unknowns to one that only contains a single unknown surface source. This reduction now allows us to have a solvable equation. Performing these operations, we get

$$E_z^{\text{inc}}(\boldsymbol{\rho}) - jk\eta \int_S g(\boldsymbol{\rho}, \boldsymbol{\rho}') J_{s,z}(\boldsymbol{\rho}') dS' = 0, \qquad \boldsymbol{\rho} \in S, \tag{4.50}$$

which is known as the *electric field integral equation (EFIE)* because it is formulated in terms of the electric field. Note that we no longer denote this integral as a principal value integral because this particular term does not require this special treatment (the singularity must still be handled carefully in numerical integration routines, but the result is more regular and does not contribute a non-zero value like the normal derivative of the Green's function did).

From a more physical perspective, we can see that this integral equation is enforcing the Dirichlet boundary on the total E_z by recognizing that the scattered field is

$$E_z^{\text{sc}} = -jk\eta \int_S g(\boldsymbol{\rho}, \boldsymbol{\rho}') J_{s,z}(\boldsymbol{\rho}') dS. \tag{4.51}$$

We can then think of our decomposition of the total field into $E_z = E_z^{\text{inc}} + E_z^{\text{sc}}$ to see that (4.50) is simply enforcing that $E_z = 0$ on S.

We may now go about solving the EFIE using the MoM. To do this, we first divide the surface of the conducting cylinder S up into a number of small segments that we can define simple basis functions over. For this particular problem, we will assume that the surface current density is constant over each segment so that we may use a pulse basis function (piecewise constant). If we then perform a point matching procedure (testing with a delta function) we will arrive at a matrix equation of

$$[Z]\{J\} = \{V\}, \tag{4.52}$$

where

$$[Z]_{mn} = jk\eta \int_{S_n} g(\boldsymbol{\rho}_m, \boldsymbol{\rho}') dS' \tag{4.53}$$

and S_n is the nth segment of the mesh and $\boldsymbol{\rho}_m$ is the center of the mth segment. We further have our excitation vector being given by

$$\{V\}_m = E_z^{\text{inc}}(\boldsymbol{\rho}_m). \tag{4.54}$$

To actually evaluate the integrals in (4.53), we need to consider the two cases of when $m = n$ and $m \neq n$ separately. The simpler case is when $m \neq n$ since we do not need to directly integrate the singularity. For this case, we can use the midpoint integration rule quite readily. When $m = n$ we can utilize the small-argument approximation for the Hankel function to determine the result of the integral. These two results are summarized as

$$[Z]_{mn} = \begin{cases} \dfrac{k\eta S_n}{4} H_0^{(2)}(k|\boldsymbol{\rho}_m - \boldsymbol{\rho}_n|), & m \neq n, \\ \dfrac{k\eta S_n}{4}\left[1 - j\dfrac{2}{\pi}\ln\left(\dfrac{k\gamma S_n}{4e}\right)\right], & m = n, \end{cases} \tag{4.55}$$

where $e \approx 2.7183$ and $\gamma \approx 1.781$.

You will note that we have somewhat suggestively written our matrix equation using notation that harkens back to Ohm's law. This is common in the CEM literature, because we can typically view the MoM matrix as being like a kind of impedance matrix that translates induced currents into the fields that they produce. Unfortunately, this notation is typically used for most electromagnetic integral equations *even when this notion no longer applies as explicitly.*

As a final note, we recall that once we have computed the solution to the EFIE we can use the results to compute the total field at any other location. We can do this using Huygens' principle specialized to our particular case of interest. Namely, we will get that

$$E_z(\boldsymbol{\rho}) = E_z^{\text{inc}}(\boldsymbol{\rho}) - jk\eta \oint_S g(\boldsymbol{\rho}, \boldsymbol{\rho}') J_{s,z}(\boldsymbol{\rho}') dS'. \tag{4.56}$$

To compute far-field results, we can utilize standard approximations for the asymptotic form of the Hankel function for large arguments to simplify the numerical integration that is needed.

4.4 2D Magnetic Field Integral Equation (MFIE)

We will now look at an alternative way to use (4.45) to formulate an integral equation for a particular problem of interest. In particular, we will consider the TE$_z$ polarization scattering from a conducting cylinder of arbitrary cross section. For this problem, the Helmholtz equation that needs to be satisfied for the TE$_z$ polarization is

$$\nabla^2 H_z(\boldsymbol{\rho}) + k^2 H_z(\boldsymbol{\rho}) = -[\nabla \times \mathbf{J}_i(\boldsymbol{\rho})]_z, \qquad \boldsymbol{\rho} \in V, \tag{4.57}$$

where $[\nabla \times \mathbf{J}_i(\boldsymbol{\rho})]_z$ is the z-component of the curl of the impressed current source which will produce the incident field for the scattering problem. The boundary conditions that must be satisfied on the surface of the cylinder are that

$$H_z(\boldsymbol{\rho}) = -J_{s,t}(\boldsymbol{\rho}), \qquad \boldsymbol{\rho} \in S, \tag{4.58}$$

$$\partial_n H_z(\boldsymbol{\rho}) = 0, \qquad \boldsymbol{\rho} \in S, \tag{4.59}$$

where $J_{s,t}$ is the surface current density tangential to the surface of the cylinder that is induced due to the incident field.

If we apply these boundary conditions in (4.45), we get

$$-\frac{1}{2}J_{s,t}(\boldsymbol{\rho}) + \fint \partial_{n'} g(\boldsymbol{\rho}, \boldsymbol{\rho}')J_{s,t}(\boldsymbol{\rho}')dS' = H_z^{\text{inc}}(\boldsymbol{\rho}), \qquad \boldsymbol{\rho} \in S, \tag{4.60}$$

which is known as the *magnetic field integral equation (MFIE)* because it has been formulated in terms of the magnetic field. Another note on terminology, we typically refer to this kind of integral equation as being an *integral equation of the second kind*. The reason for this is because the unknown that we are attempting to solve for, $J_{s,t}$ in this case, appears both inside and outside of the integral. This is in contrast to the EFIE, where the unknown to be solved for was purely inside the integral of the equation. The EFIE is known as an *integral equation of the first kind* in this terminology. These two "kinds" of integral equations are separated by this notation because they have very different numerical properties that are important to consider in a more detailed application of the MoM.

We can now solve (4.60) using the same approach that we utilized for the EFIE. Our matrix equation has the same form as that of (4.52), but the individual elements of the matrices and vectors will change. In particular, we will have that

$$[Z]_{mn} = -\frac{1}{2}\delta_{mn} + \fint_{S_n} \partial_{n'} g(\boldsymbol{\rho}_m, \boldsymbol{\rho}')dS', \tag{4.61}$$

where δ_{mn} is the Kronecker delta function, and the excitation vector becomes

$$\{V\}_m = H_z^{\text{inc}}(\boldsymbol{\rho}_m). \tag{4.62}$$

Taking the derivative of the Green's function in (4.61) requires us to be careful about our approach. We can evaluate this using properties of the Hankel functions and basis vector calculus to get

$$\begin{aligned} \partial_{n'} g(\boldsymbol{\rho}_m, \boldsymbol{\rho}') &= \frac{1}{4j}\hat{n}' \cdot \nabla' H_0^{(2)}(k|\boldsymbol{\rho}_m - \boldsymbol{\rho}'|) \\ &= -\frac{k}{4j}H_1^{(2)}(k|\boldsymbol{\rho}_m - \boldsymbol{\rho}'|)\, \hat{n}' \cdot \nabla'|\boldsymbol{\rho}_m - \boldsymbol{\rho}'| \\ &= \frac{k}{4j}H_1^{(2)}(k|\boldsymbol{\rho}_m - \boldsymbol{\rho}'|)\, \frac{\hat{n}' \cdot (\boldsymbol{\rho}_m - \boldsymbol{\rho}')}{|\boldsymbol{\rho}_m - \boldsymbol{\rho}'|}. \end{aligned} \tag{4.63}$$

Using the midpoint integration we can then find that the matrix elements will be

$$[Z]_{mn} = \begin{cases} \dfrac{kS_n}{4j}H_1^{(2)}(k|\boldsymbol{\rho}_m - \boldsymbol{\rho}_n|)\, \dfrac{\hat{n}' \cdot (\boldsymbol{\rho}_m - \boldsymbol{\rho}_n)}{|\boldsymbol{\rho}_m - \boldsymbol{\rho}_n|}, & m \neq n, \\ -\dfrac{1}{2}, & m = n. \end{cases} \tag{4.64}$$

Results from the EFIE and MFIE are shown in Fig. 4.4 for the scattering from a circular cylinder with radius of 1λ. Due to the symmetrical shape, an analytical solution can be used to validate the MoM results. This kind of analytical solution is typically referred to as a *Mie series solution*, although this is more common terminology for the corresponding solution to scattering from a sphere.

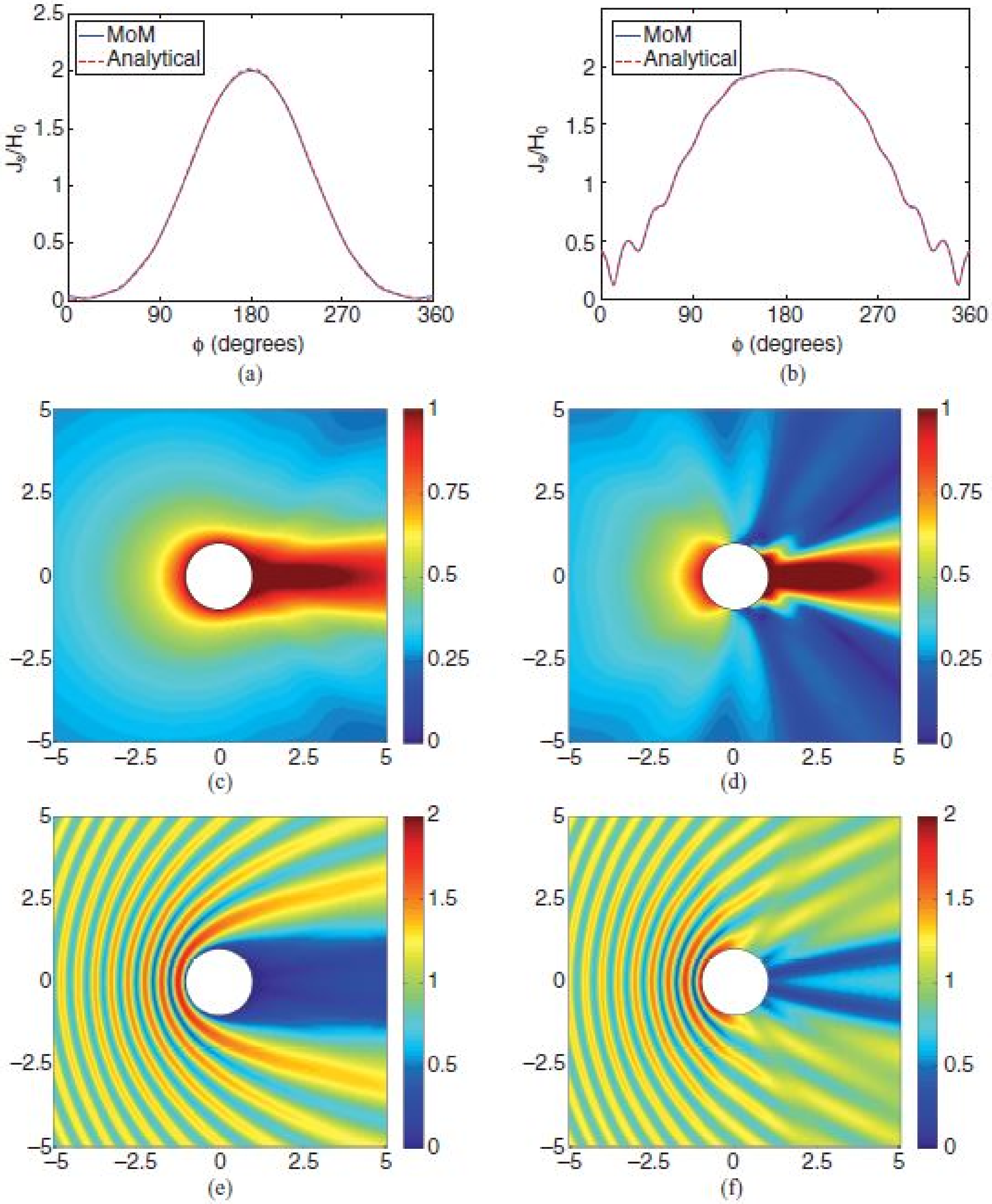


Figure 10.5 Scattering by a circular conducting cylinder with a radius of 1λ. The incident wave propagates from the left to right. The left and right columns show the results for the TM and TE polarizations, respectively. (a) Magnitude of the induced surface current density $J_{s,z}$. (b) Magnitude of the induced surface current density $J_{s,t}$. (c) Magnitude of the scattered field E_z^{sc}. (d) Magnitude of the scattered field H_z^{sc}. (e) Magnitude of the total field E_z. (f) Magnitude of the total field H_z. The values of the fields are normalized by the magnitude of their respective incident fields.

Figure 4.4: Results for scattering from a conducting cylinder using the EFIE and MFIE (images from [5]).

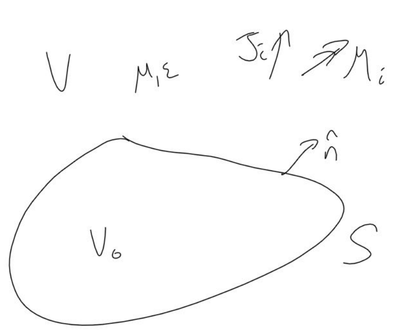


Figure 4.5: Problem illustration for a generic 3D scattering problem that integral equations can be readily formulated for.

4.5 Formulation of Integral Equations for 3D Wave Equation

As mentioned previously, there are a number of different ways to go about deriving integral equations that are all more or less equivalent. However, some approaches are more "general" and can suggest ways to formulate less "standard" or more "exotic" kinds of integral equations. We will now follow one of these more general approaches to derive integral equations for the 3D Helmholtz equation. Our particular approach will try to be more physically intuitive by making use of the surface equivalence principle you learned about in ECE 604, although the general techniques can be justified through purely rigorous mathematics as well [30].

To formulate an integral equation, we will consider the problem illustrated in Fig. 4.5. Here, we have an arbitrary source distribution that produces some incident electric and magnetic fields, denoted by $\mathbf{E}^{inc}$ and $\mathbf{H}^{inc}$, respectively. These source distributions are located in a homogeneous region external to our object of interest, which is specified by the surface S. Just like the 2D case, we will not worry about the specific properties within the region of interest initially since taking them into account will be part of specializing our general equations to a particular problem of interest.

At a high level, our goal will be to utilize the surface equivalence principle to arrive at an equivalent problem that involves a set of surface sources radiating in a completely homogeneous medium. When this is the case, we can directly integrate the source distributions with the homogeneous medium Green's function to calculate the electromagnetic potentials anywhere in space. By then taking appropriate derivatives of these expressions, we will have an expression for how to calculate the fields produced by the surface sources. We can then take the limit of these expressions as the observation point approaches the surface S to derive a general integral equation that can be further specialized to particular cases of interest (e.g., a conducting surface or a penetrable scatterer). With this in mind, we will now begin to review a few key points from electromagnetic theory that will be needed in this overall derivation process.

4.5.1 Potentials Produced by Known Source Distributions

When we first started to discuss electromagnetic integral equations we spent some time considering how one of the convenient properties of a Green's function was that it could be used to invert a partial differential equation. For an arbitrary problem (e.g., inhomogeneous medium), it is often extremely difficult to analytically determine the Green's function for the problem. However, if we have a particularly simple problem like the wave equation in a homogeneous region, it becomes tractable to determine the Green's function. You may recall from your previous electromagnetic theory courses that the Green's function can be most easily used in this way to compute the electromagnetic *potentials* rather than the fields. To see this, we will briefly recall some basic points about the electromagnetic potentials.

To begin, we will assume that we are dealing with the form of Maxwell's equations that only contain electric sources (e.g., electric current and electric charge densities); i.e.,

$$\nabla \times \mathbf{H} = j\omega \mathbf{D} + \mathbf{J}, \tag{4.65}$$

$$\nabla \times \mathbf{E} = -j\omega \mathbf{B}, \tag{4.66}$$

$$\nabla \cdot \mathbf{D} = \rho, \tag{4.67}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{4.68}$$

We can simplify this problem from attempting to solve for 4 vectors to the problem of solving for 1 scalar and 1 vector unknown through the use of the potentials. These are introduced by first noting that because of (4.68), we can always express $\mathbf{B}$ as the curl of some vector as

$$\mathbf{B} = \nabla \times \mathbf{A}, \tag{4.69}$$

where $\mathbf{A}$ is known as the *magnetic vector potential.* This can be substituted into (4.66), and after consolidating all terms under the curl operation we get that

$$\nabla \times \big(\mathbf{E} + j\omega \mathbf{A}\big) = 0. \tag{4.70}$$

We can immediately recognize that this will always be satisfied if we express $\mathbf{E} + j\omega\mathbf{A}$ as the gradient of some scalar as

$$\mathbf{E} + j\omega \mathbf{A} = -\nabla \Phi, \tag{4.71}$$

where Φ is the *electric scalar potential.* We typically rearrange this to express $\mathbf{E}$ purely in terms of the potentials as

$$\mathbf{E} = -j\omega \mathbf{A} - \nabla \Phi. \tag{4.72}$$

Next, we use these potentials in Ampere's law to determine wave equations for them. In a homogeneous medium, Ampere's law is

$$\nabla \times \mu^{-1} \mathbf{B} = j\omega\epsilon \mathbf{E} + \mathbf{J}. \tag{4.73}$$

We can substitute in our potentials using (4.69) and (4.72) to get

$$\nabla \times \mu^{-1} \nabla \times \mathbf{A} = \epsilon\big(\omega^2 \mathbf{A} - j\omega \nabla \Phi\big) + \mathbf{J}. \tag{4.74}$$

We can rearrange terms to get

$$\nabla \times \nabla \times \mathbf{A} - k^2 \mathbf{A} = -\mu\epsilon j\omega \nabla \Phi + \mu \mathbf{J}. \tag{4.75}$$

We see that the left-hand side looks like one of our familiar wave equations with a propagation speed matching the speed of light.

We may now use our standard vector identity to rewrite the $\nabla \times \nabla \times$ operator to get

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = \nabla\big(\nabla \cdot \mathbf{A}\big) + \mu\epsilon j\omega \nabla \Phi - \mu \mathbf{J}. \tag{4.76}$$

Have we accomplished anything? Not yet, but if you recall the Helmholtz decomposition theorem, we realize that we have some freedom in specifying what $\nabla \cdot \mathbf{A}$ should be equal to. This freedom exists because we have introduced these auxiliary potential functions to help us solve our problem, they do not already come fully specified like our field and fluxes did. It is this freedom that we can exploit to help us simplify the solution of certain problems.

To move forward, we need to specify what we are going to force $\nabla \cdot \mathbf{A}$ to equal. This is called setting a *gauge condition*. Although electromagnetic theory was one of the first areas where gauge conditions were widely used, it has become a very prevalent concept in many areas of modern physics to describe different kinds of forces. For instance, gauge conditions are a very prevalent and important part of the Standard Model of particle physics, which in addition to electromagnetism includes theories for the weak and strong nuclear forces. These additional forces obey a set of equations that can be viewed as a generalization of Maxwell's equations (albeit, this takes some fairly sophisticated mathematics to see).

For our current purposes, let us use our gauge condition to try and simplify our wave equation. We can do this by making the two terms involving $\mathbf{A}$ and Φ cancel on the right-hand side of (4.76). That is,

$$\nabla \cdot \mathbf{A} = -\mu\epsilon j\omega \Phi. \tag{4.77}$$

This gauge condition is used frequently enough that it has its own name. It is the *Lorenz gauge condition*. Note that many people erroneously will call this the Lorentz gauge condition. This is incorrect and should not be done. Both Lorenz and Lorentz made important contributions to electromagnetic theory, and they should be appropriately commended for their contributions.

Now, after setting this gauge condition, our vector potential wave equation in the Lorenz gauge becomes

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = -\mu \mathbf{J}. \tag{4.78}$$

What does the equation for Φ look like? We can find this out by substituting our electromagnetic potentials into Gauss' law of electricity. This takes us from having

$$\nabla \cdot \mathbf{E} = \rho/\epsilon, \tag{4.79}$$

to having

$$\nabla \cdot \big(\nabla\Phi + j\omega\mathbf{A}\big) = -\rho/\epsilon. \tag{4.80}$$

We can use the Lorenz gauge condition to rewrite this as

$$\nabla^2\Phi + k^2\Phi = -\rho/\epsilon. \tag{4.81}$$

We see that we get a *scalar Helmholtz equation* for Φ. This is one of the advantages of the Lorenz gauge. It allows us to arrive at decoupled wave equations for our potentials that immediately show that they both propagate at the speed of light. This is part of the reason why this gauge condition is frequently utilized in the study of special relativity.

A similar process may be repeated for a set of Maxwell's equations that only consider magnetic sources (i.e., magnetic current and magnetic charge densities). In this process, we need to introduce an *electric vector potential* $\mathbf{F}$ and a *magnetic scalar potential* Φ_m. We can eventually show that if we enforce a Lorenz gauge between these potentials as well, then these potentials satisfy Helmholtz wave equations that are

$$\nabla^2\mathbf{F} + k^2\mathbf{F} = -\epsilon\mathbf{M}, \tag{4.82}$$

$$\nabla^2\Phi_m + k^2\Phi_m = -\rho_m/\mu, \tag{4.83}$$

where $\mathbf{M}$ is a magnetic current density and ρ_m is a magnetic charge density. Note that in terms of all *four* of these potentials, we may finally express the electric and magnetic fields as

$$\mathbf{E} = -j\omega\mathbf{A} - \nabla\Phi - \epsilon^{-1}\nabla\times\mathbf{F}, \tag{4.84}$$

$$\mathbf{H} = \mu^{-1}\nabla\times\mathbf{A} - j\omega\mathbf{F} - \nabla\Phi_m. \tag{4.85}$$

From this process, we see that all of the potentials satisfy a simple Helmholtz wave equation. If we are in a homogeneous medium, we may then use the Green's function that satisfies

$$\nabla^2 g(\mathbf{r},\mathbf{r}') + k^2 g(\mathbf{r},\mathbf{r}') = -\delta(\mathbf{r}-\mathbf{r}') \tag{4.86}$$

to invert *all* of these wave equations. The particular Green's function that satisfies (4.86) has the well-known expression of

$$g(\mathbf{r},\mathbf{r}') = \frac{e^{-jkR}}{4\pi R}, \tag{4.87}$$

where $R = |\mathbf{r}-\mathbf{r}'|$. We can use this to express each of the potentials as a convolution of the Green's function against the particular source distribution. This allows us to compute the potentials as

$$\Phi(\mathbf{r}) = \epsilon^{-1}\iiint \rho(\mathbf{r}')\frac{e^{-jkR}}{4\pi R}dV', \tag{4.88}$$

$$\Phi_m(\mathbf{r}) = \mu^{-1} \iiint \rho_m(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV', \tag{4.89}$$

$$\mathbf{A}(\mathbf{r}) = \mu \iiint \mathbf{J}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV', \tag{4.90}$$

$$\mathbf{F}(\mathbf{r}) = \epsilon \iiint \mathbf{M}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV'. \tag{4.91}$$

We may use these expressions of the potentials in (4.84) and (4.85) to find expressions for the electric and magnetic fields in terms of the source distributions. For the electric field, we get

$$\mathbf{E}(\mathbf{r}) = -\iiint j\omega\mu\mathbf{J}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV' - \nabla \iiint \epsilon^{-1}\rho(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV' - \nabla \times \iiint \mathbf{M}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV'. \tag{4.92}$$

We can utilize the current continuity equation to rewrite the charge density into a current density so that (4.92) only involves two source distributions (the current densities). This gives

$$\begin{aligned}\mathbf{E}(\mathbf{r}) = -\iiint j\omega\mu\mathbf{J}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV' + \nabla \iiint (j\omega\epsilon)^{-1} [\nabla' \cdot \mathbf{J}(\mathbf{r}')] \frac{e^{-jkR}}{4\pi R} dV' \\ - \nabla \times \iiint \mathbf{M}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV'.\end{aligned} \tag{4.93}$$

A similar process can be completed to write the magnetic field as

$$\begin{aligned}\mathbf{H}(\mathbf{r}) = \nabla \times \iiint \mathbf{J}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} - \iiint j\omega\epsilon\mathbf{M}(\mathbf{r}') \frac{e^{-jkR}}{4\pi R} dV' \\ + \nabla \iiint (j\omega\mu)^{-1} [\nabla' \cdot \mathbf{M}(\mathbf{r}')] \frac{e^{-jkR}}{4\pi R} dV'.\end{aligned} \tag{4.94}$$

We will be able to use these expressions for **E** and **H** in conjunction with the surface equivalence principle to derive integral equations. However, it will first be helpful to review some aspects about the surface equivalence principle before considering the formulation of the complete integral equations.

4.5.2 Surface Equivalence Principle Review

To illustrate a simple example of the surface equivalence principle, we will consider the problem shown in Fig. 4.6. Here, we have a set of sources that produce a set of fields **E** and **H** throughout all space. We draw a "mathematical surface" S around the sources to break our description of the problem into two regions. If we are only interested in the fields outside of the surface S, we can modify the interior fields to be different values $\mathbf{E}'$ and $\mathbf{H}'$

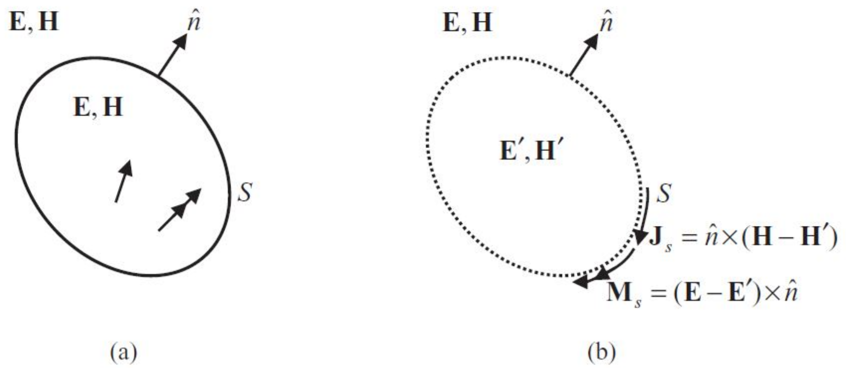

E, H
$\hat{n}$
E, H
S
(a)
E, H
$\hat{n}$
E′, H′
S
$\mathbf{J}_s = \hat{n} \times (\mathbf{H} - \mathbf{H}')$
$\mathbf{M}_s = (\mathbf{E} - \mathbf{E}') \times \hat{n}$
(b)

Figure 4.6: Illustration of the surface equivalence principle. (a) The original problem of a set of sources inside a "mathematical surface" and (b) the equivalent problem for the exterior field (images from [5]).

to assist in simplifying a particular problem. Since there is now a "jump" or discontinuity in the values of tangential fields to S, this must be compensated for by the presence of a set of surface currents according to the boundary conditions for Maxwell's equations. These equivalent currents will produce the tangential fields $\hat{n} \times \mathbf{E}$ and $\hat{n} \times \mathbf{H}$ just outside S and $\hat{n} \times \mathbf{E}'$ and $\hat{n} \times \mathbf{H}'$ just inside S. Now, because these tangential fields are specified over an entire closed surface, we know from the uniqueness theorem that in the exterior region these equivalent currents will produce the same fields as in the original problem regardless of how we pick $\mathbf{E}'$ and $\mathbf{H}'$.

Hence, we can use this degree of freedom to simplify our problem setup. A common example of this would be to set $\mathbf{E}' = \mathbf{H}' = 0$ so that no fields exist within S. Since there are no fields there anymore, we can change the problem arbitrarily by modifying the "material" in the region. Hence, if S were marking the surface of some inhomogeneity in an otherwise homogeneous background medium we could use the equivalence principle to change the inhomogeneity to the same properties as the background material. With a completely homogeneous region, the equivalent surface currents can then be integrated against the homogeneous medium Green's functions in the manner shown previously to determine the fields at any location outside of the surface. We will now use this basic process to formulate a surface integral equation through use of an appropriately defined equivalent problem.

4.5.3 Integral Equation Formulation

The problem we wish to formulate an integral equation for is shown in Fig. 4.7. We have a set of sources that produce incident fields that would exist throughout all of space in the absence of the inhomogeneity with surface S. To solve the complete problem, a set of scattered fields will need to be produced both exterior and interior to S so that Maxwell's equations are correctly satisfied in all regions of the problem. However, if we begin by only considering deriving an integral equation for the exterior region we can develop a suitable equivalent problem to allow us to use all the machinery we have covered in earlier sections.

In particular, we will define a set of equivalent currents on S that produce $\mathbf{E}^{sc,1}$ and $\mathbf{H}^{sc,1}$

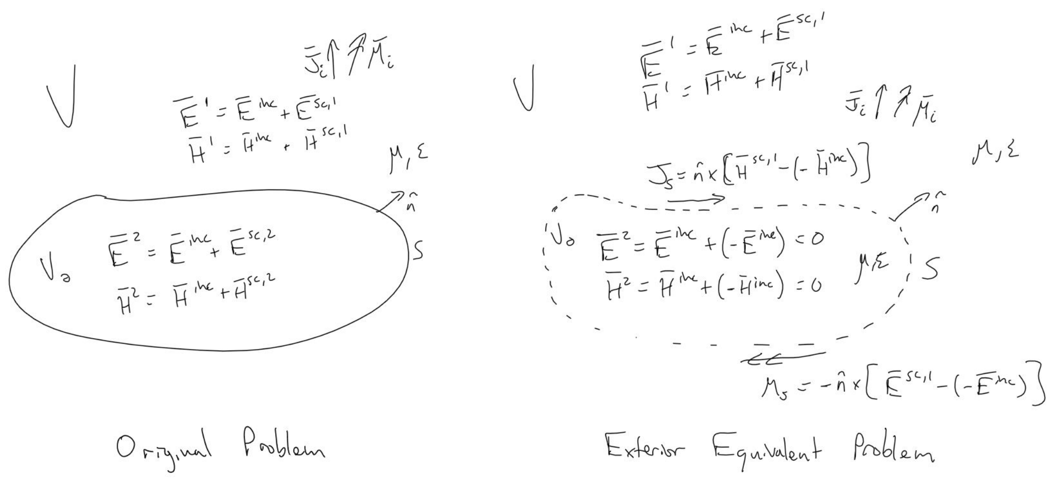

$\bar{J}_i$ $\bar{M}_i$
V
$\bar{E}^1 = \bar{E}^{inc} + \bar{E}^{sc,1}$
$\bar{H}^1 = \bar{H}^{inc} + \bar{H}^{sc,1}$
μ, ε
$\hat{n}$
V_o
$\bar{E}^2 = \bar{E}^{inc} + \bar{E}^{sc,2}$
$\bar{H}^2 = \bar{H}^{inc} + \bar{H}^{sc,2}$
S
Original Problem
V
$\bar{E}^1 = \bar{E}^{inc} + \bar{E}^{sc,1}$
$\bar{H}^1 = \bar{H}^{inc} + \bar{H}^{sc,1}$
$\bar{J}_i$ $\bar{M}_i$
μ, ε
$J_s = \hat{n} \times [\bar{H}^{sc,1} - (-\bar{H}^{inc})]$
$\hat{n}$
V_o
$\bar{E}^2 = \bar{E}^{inc} + (-\bar{E}^{inc}) = 0$
$\bar{H}^2 = \bar{H}^{inc} + (-\bar{H}^{inc}) = 0$
μ, ε
S
$M_s = -\hat{n} \times [\bar{E}^{sc,1} - (-\bar{E}^{inc})]$
Exterior Equivalent Problem

Figure 4.7: Illustration of the formulation of an integral equation through the surface equivalence principle. (left) The original problem and (right) the equivalent exterior problem.

in the exterior region and $-\mathbf{E}^{inc}$ and $-\mathbf{H}^{inc}$ in the interior region. Hence, we have that the equivalent currents are

$$\mathbf{J} = \hat{n} \times \left[\mathbf{H}^{sc,1} - (-\mathbf{H}^{inc})\right] = \hat{n} \times \mathbf{H}^1, \tag{4.95}$$

$$\mathbf{M} = -\hat{n} \times \left[\mathbf{E}^{sc,1} - (-\mathbf{E}^{inc})\right] = \hat{n} \times \mathbf{E}^1. \tag{4.96}$$

We see that they are equal to the tangential components of the total field in the exterior region. This will be useful when it comes time to utilize boundary conditions in the formulation of specific integral equations for a problem such as when S marks the boundary of a PEC inhomogeneity.

The other important aspect of defining the equivalent currents in this way is that they force the field internal to S to be identically 0. This is another statement of the *extinction theorem* that we introduced in a different context within the formulation of 2D integral equations. Now, because the field is 0 inside S we can follow our "standard" surface equivalence principle approach and replace the material that was here with the same background material that exists in the homogeneous region exterior to S. We then have a set of surface sources that radiate in a completely homogeneous medium. Hence, we can utilize the integral representations of

$$\mathbf{E}(\mathbf{r}) = -\iiint j\omega\mu\mathbf{J}(\mathbf{r}')\frac{e^{-jkR}}{4\pi R}dV' + \nabla\iiint (j\omega\epsilon)^{-1}[\nabla'\cdot\mathbf{J}(\mathbf{r}')]\frac{e^{-jkR}}{4\pi R}dV' \\ - \nabla\times\iiint \mathbf{M}(\mathbf{r}')\frac{e^{-jkR}}{4\pi R}dV' \tag{4.97}$$

and

$$\mathbf{H}(\mathbf{r}) = \nabla \times \iiint \mathbf{J}(\mathbf{r}')\frac{e^{-jkR}}{4\pi R} - \iiint j\omega\epsilon\mathbf{M}(\mathbf{r}')\frac{e^{-jkR}}{4\pi R}dV' \\ + \nabla \iiint (j\omega\mu)^{-1}[\nabla'\cdot\mathbf{M}(\mathbf{r}')]\frac{e^{-jkR}}{4\pi R}dV' \quad (4.98)$$

to compute the electric and magnetic fields anywhere. We will get that

$$\iint \left[- j\omega\mu g(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}') + \nabla g(\mathbf{r},\mathbf{r}')\frac{\nabla'\cdot\mathbf{J}(\mathbf{r}')}{j\omega\epsilon}\right]dS' \\ - \nabla \times \iint g(\mathbf{r},\mathbf{r}')\mathbf{M}(\mathbf{r}')dS' = \begin{cases}\mathbf{E}^{sc,1}(\mathbf{r}), & \mathbf{r}\in V,\\ -\mathbf{E}^{inc}(\mathbf{r}), & \mathbf{r}\in V_0,\end{cases} \quad (4.99)$$

$$\iint \left[- j\omega\epsilon g(\mathbf{r},\mathbf{r}')\mathbf{M}(\mathbf{r}') + \nabla g(\mathbf{r},\mathbf{r}')\frac{\nabla'\cdot\mathbf{M}(\mathbf{r}')}{j\omega\mu}\right]dS' \\ + \nabla \times \iint g(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')dS' = \begin{cases}\mathbf{H}^{sc,1}(\mathbf{r}), & \mathbf{r}\in V,\\ -\mathbf{H}^{inc}(\mathbf{r}), & \mathbf{r}\in V_0.\end{cases} \quad (4.100)$$

We can derive integral equations by taking the limit of these expressions as $\mathbf{r} \to S$.

4.5.4 Bringing r to S

We will now look at how to go about bringing the observation point to the surface S. However, before doing this it will be useful for us to scale some of our quantities so that we can write (4.99) and (4.100) in a more consistent form. In particular, if we define a scaled electric current density and magnetic field as

$$\overline{\mathbf{J}} = \eta\mathbf{J} \quad (4.101)$$

$$\overline{\mathbf{H}} = \eta\mathbf{H}, \quad (4.102)$$

we can rewrite our integral representations as

$$-\mathcal{L}\{\,\overline{\mathbf{J}}\,\} + \mathcal{K}\{\mathbf{M}\} = \begin{cases}\mathbf{E}^{sc,1}(\mathbf{r}), & \mathbf{r}\in V,\\ -\mathbf{E}^{inc}(\mathbf{r}), & \mathbf{r}\in V_0,\end{cases} \quad (4.103)$$

$$-\mathcal{L}\{\mathbf{M}\} - \mathcal{K}\{\,\overline{\mathbf{J}}\,\} = \begin{cases}\overline{\mathbf{H}}^{sc,1}(\mathbf{r}), & \mathbf{r}\in V,\\ -\overline{\mathbf{H}}^{inc}(\mathbf{r}), & \mathbf{r}\in V_0,\end{cases} \quad (4.104)$$

where

$$\mathcal{L}\{\mathbf{X}\} = \iint_S \left[g(\mathbf{r},\mathbf{r}')jk\mathbf{X}(\mathbf{r}') - \nabla g(\mathbf{r},\mathbf{r}')\frac{\nabla'\cdot\mathbf{X}(\mathbf{r}')}{jk}\right]dS' \quad (4.105)$$

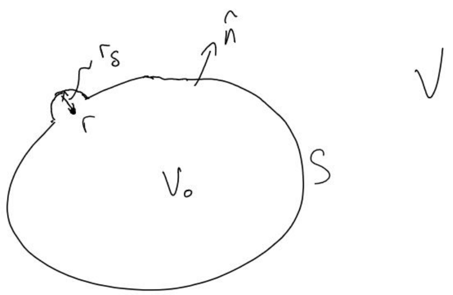


Figure 4.8: Deformation of the integration contour for evaluating the principal value of the 3D integral operators relevant to electromagnetic integral equations.

$$\mathcal{K}\{\mathbf{X}\} = \iint_S \mathbf{X}(\mathbf{r}') \times \nabla g(\mathbf{r}, \mathbf{r}') dS'. \tag{4.106}$$

We can now see that all we need to consider for bringing $\mathbf{r} \to S$ is how the two *integral operators* defined in (4.105) and (4.106) behave as we take this limit. To do this, we will follow a similar process to what we did in the 2D case. That is, we will deform the surface slightly around the singular point and then take the limit as the deformation shrinks to 0. Since we are doing a 3D problem now, our surface will need to be deformed using a hemispherical path. If we take this limit with $\mathbf{r} \in V_0$, we will need to deform the surface so that the hemisphere "sticks out" of the surface into the exterior region as shown in Fig. 4.8.

As a final point, because we are dealing with vector fields we will need to consider the different scalar components carefully since they can behave differently. In general, the two tangential components will behave similarly, but this behavior may be distinct compared to the normal component. For most standard electromagnetic integral equations, we only need to care about the behavior of the tangential components so we will only focus on this here. To do this, we will take the cross product of the integral operators with the normal vector at the observation point as we take the limit.

When this analysis is done, it is found that the singular term does not contribute to the evaluation of $\hat{n} \times \mathcal{L}\{\mathbf{X}\}$ so that it may be "ignored" (special care is still needed in handling the integrals numerically in the MoM, but there is no principal value term that needs to be explicitly extracted). In contrast to this, the singular term of $\hat{n} \times \mathcal{K}\{\mathbf{X}\}$ does not vanish. To evaluate this contribution, it is easiest to recall that the different equivalent currents are actually related to the cross product of an electric or magnetic field so we will rewrite $\mathbf{X}$ as

$\mathbf{X}(\mathbf{r}') = \hat{n}' \times \mathbf{Y}(\mathbf{r}')$. Using this, we get that the singular contribution can be evaluated as

$$\begin{aligned}
\hat{n} \times \int_0^{2\pi} \int_0^{\pi/2} \mathbf{X} \times \nabla g(\mathbf{r}, \mathbf{r}') r_\delta^2 \sin\theta d\theta d\phi &= -\hat{n} \times \int_0^{2\pi} \int_0^{\pi/2} \mathbf{X} \times \nabla' g(\mathbf{r}, \mathbf{r}') r_\delta^2 \sin\theta d\theta d\phi \\
&= \hat{n} \times \int_0^{2\pi} \int_0^{\pi/2} (\hat{n}' \times \mathbf{Y}) \times \hat{r}' \frac{e^{-jkr_\delta}}{4\pi} \sin\theta d\theta d\phi \\
&= \hat{n} \times \int_0^{2\pi} \int_0^{\pi/2} (\hat{n}' \cdot \hat{r}') \mathbf{Y} \frac{e^{-jkr_\delta}}{4\pi} \sin\theta d\theta d\phi \\
&= \hat{n} \times \int_0^{2\pi} \int_0^{\pi/2} \mathbf{Y} \frac{e^{-jkr_\delta}}{4\pi} \sin\theta d\theta d\phi \\
&= \frac{1}{2}\mathbf{X}, \qquad r_\delta \to 0,
\end{aligned} \tag{4.107}$$

where $\hat{r}'$ is the unit normal vector on the hemispherical surface (which naturally matches $\hat{n}$ in this case).

Using this result, we can now write our integral representations on the actual surface S for the case of the exterior equivalent problem. Our end result is that

$$\frac{1}{2}\mathbf{M} + \hat{n} \times \tilde{\mathcal{K}}\{\mathbf{M}\} - \hat{n} \times \mathcal{L}\{\,\overline{\mathbf{J}}\,\} = -\hat{n} \times \mathbf{E}^{inc}(\mathbf{r}), \quad \mathbf{r} \in S_-, \tag{4.108}$$

$$\frac{1}{2}\overline{\mathbf{J}} + \hat{n} \times \tilde{\mathcal{K}}\{\,\overline{\mathbf{J}}\,\} + \hat{n} \times \mathcal{L}\{\mathbf{M}\} = \hat{n} \times \overline{\mathbf{H}}^{inc}(\mathbf{r}), \quad \mathbf{r} \in S_-, \tag{4.109}$$

where

$$\tilde{\mathcal{K}}\{\mathbf{X}\} = \text{P.V.} \iint_S \mathbf{X}(\mathbf{r}') \times \nabla g(\mathbf{r}, \mathbf{r}') dS' \tag{4.110}$$

is the $\mathcal{K}$-operator with the singular point excluded. We have also labeled these equations as being valid on S_- to emphasize that they were derived for Fig. 4.8 with the observation point located in V_0. We can now utilize these equations to derive some of the most commonly used integral equations for the 3D analysis of electromagnetic systems.

4.6 Basis Functions for Surface Integral Equations

Before we discuss the particular integral equations that are most commonly used, it will be important for us to determine suitable basis and testing functions to use in the analysis. We will primarily focus on a single function, known as the Rao-Wilton Glisson (RWG) function [31]. This was the primary function used in the CEM literature for an extended period of time. Eventually, a more rigorous analysis of the underlying mathematical theory of the integral equations being solved highlighted that this function leads to poorer performance for certain integral equations if it is used both as the basis and testing functions [16]. After this realization, additional functions were determined that helped resolve these issues (although

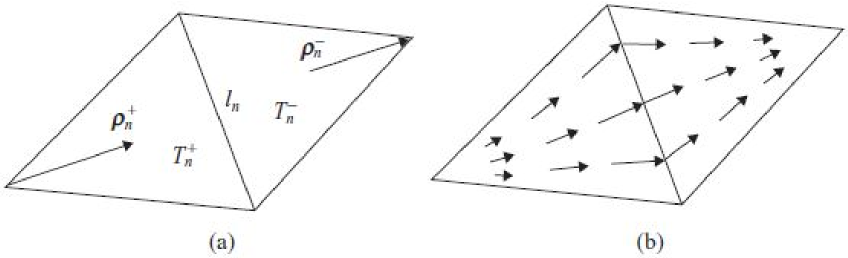


Figure 4.9: (a) Illustration of the quantities involved in the definition of the RWG function and (b) vector plot of the RWG function over the two joined triangles (images from [5]).

not to a completely satisfactory extent, so this is still an ongoing area of research interest) [32].

Now, returning to the RWG function, we first must decide on a suitable mesh to use for an arbitrary surface. The simplest shapes that can do a reasonably good job modeling most surfaces are triangular patches. Considering this, we will first divide the entire surface S into small triangular patches in a manner similar to FEM (i.e., we keep the patches small so that a simple basis function can faithfully represent the surface current over the extent of the patch). We can then define a RWG function to be associated with each *interior edge* of the surface mesh, with the function spanning the two triangles that share the interior edge. If we define for the nth edge one triangle as the "positive" triangle T_n^+ and the other triangle as the "negative" triangle T_n^-, the RWG function can be given as

$$\mathbf{f}_n(\mathbf{r}) = \begin{cases} \dfrac{\ell_n}{2A_n^+}\boldsymbol{\rho}_n^+, & \mathbf{r} \in T_n^+ \\ \\ \dfrac{\ell_n}{2A_n^-}\boldsymbol{\rho}_n^-, & \mathbf{r} \in T_n^-, \end{cases} \tag{4.111}$$

where ℓ_n is the length of the edge the RWG function is associated with, $A_n^\pm$ is the area of the triangles, and the definitions of $\boldsymbol{\rho}_n^\pm$ are illustrated in Fig. 4.9. These vectors point between the node of the triangle that is not attached to the edge of the RWG function and the point the RWG function is being evaluated at. To keep the vector direction of the surface current flow correct across the edge, $\boldsymbol{\rho}_n^-$ points into the unattached node of T_n^- while $\boldsymbol{\rho}_n^+$ points away from the unattached node of T_n^+.

Although this basis function has many useful properties, its most important one is that its normal component to edge ℓ_n is a constant (normalized to 1) and the normal components to all other edges are 0. This guarantees the continuity of current flow across all edges of the mesh, which is a vital property for the function to be able to accurately represent a surface current density. Inspecting (4.111), we also see that the RWG function provides a linear level of interpolation accuracy.

In contrast to FEM, it is much harder to develop suitable higher-order basis functions for use with surface integral equations. Part of the reason is that the accuracy of the solution strongly depends on the geometric fidelity of the model. If the triangular mesh cannot resolve the curvature of the surface accurately enough, having higher-order polynomial interpolation accuracy doesn't provide as significant of a boost to performance. As a result, higher-order basis functions can require the use of higher-order meshes, which tend to still be a research field of their own. There are also complications related to integrating these higher-order functions near the singularities that are present in the evaluation of the integral equations due to the Green's function. As a result, even though higher-order MoM approaches have been developed, they are not nearly as popular as higher-order FEM.

4.7 Integral Equations for 3D Conducting Geometries

We will now consider the development of three different integral equations applicable to perfectly conducting geometries embedded in a homogeneous background medium. We will devise these through specializations of (4.108) and (4.109).

To begin, we recall that the equivalent currents are equal to the tangential components of the *total fields* on the surface of the conducting geometries due to the particular surface equivalence principle formulation that we used. Hence, we can utilize the homogeneous Dirichlet boundary condition from the PEC problem we are considering to note that

$$\hat{n} \times \mathbf{E}(\mathbf{r}) = 0, \qquad \mathbf{r} \in S. \tag{4.112}$$

We can utilize this in our previous integral equation formulations to eliminate the magnetic current density unknown, since $\mathbf{M} = -\hat{n} \times \mathbf{E}$.

Now, we will make this simplification in (4.108) to get

$$\hat{n} \times \mathcal{L}\{\, \overline{\mathbf{J}}\, \} = \hat{n} \times \mathbf{E}^{inc}(\mathbf{r}), \qquad \mathbf{r} \in S. \tag{4.113}$$

Due to this equation being formulated in terms of the electric field, it is known as the *electric field integral equation (EFIE)*. We can also recognize that this is an *integral equation of the first kind* because the unknown $\overline{\mathbf{J}}$ only appears inside the integral operator. Physically, we can interpret the left-hand side as being the scattered electric field. Considering this, we can see that this equation enforces that the total tangential electric field is 0 on the surface of the PEC object (or just inside it, i.e., at S_{-}). We will consider how to solve this equation using the MoM after introducing the two other integral equations.

The second integral equation formulation comes from eliminating the magnetic current density unknowns in (4.109). This gives us

$$\frac{1}{2}\overline{\mathbf{J}} + \hat{n} \times \tilde{\mathcal{K}}\{\, \overline{\mathbf{J}}\, \} = \hat{n} \times \overline{\mathbf{H}}^{inc}(\mathbf{r}), \qquad \mathbf{r} \in S_{-}, \tag{4.114}$$

which is known as the *magnetic field integral equation (MFIE)*. We can also recognize that this is an *integral equation of the second kind* because the unknown $\overline{\mathbf{J}}$ appears both inside and outside the integral operator. Physically, we can think of this equation as enforcing that the total magnetic field is 0 just inside the PEC object at S_{-}. Note that this is *slightly* less general of a statement than what we made for the EFIE. There are consequences to this: the MFIE can only be applied to closed conductors while the EFIE can be applied to infinitely thin PEC sheets. This is not a significant issue for the MFIE since all real objects have a thickness, but there are plenty of situations where approximating an object as infinitely thin leads to acceptable numerical results and can make the total number of unknowns that need to be solved for much lower. As a result, we cannot think of the EFIE and MFIE as being completely interchangeable for the problems they can analyze.

Typically, integral equations of the second kind lead to better conditioned matrix equations that can be solved more quickly with an iterative solver than an integral equation of the first kind. However, the second-kind integral equations often achieve slightly lower accuracy than the first-kind integral equations for the same mesh resolution. This difference in accuracy can be overcome using more sophisticated discretization techniques that utilize different basis and testing functions, but this comes at the cost of an increased matrix fill time and involves a more complex code implementation for the most well-known approach in this vein [16].

Both of these integral equations suffer from the problem of *interior resonances* when they are applied to closed conductors in a lossless background medium. The issue is that the same integral equation also applies to the "interior" problem of analyzing a cavity filled with the same background material. This cavity problem supports non-trivial solutions at resonant frequencies of the cavity with no excitation applied, and hence, constitute a null space for this problem. This null space can plague the numerical solution of the "exterior problem" near the interior resonant frequencies of the cavity, leading to erroneous solutions. As the frequency of analysis grows, the number of cavity modes becomes progressively denser making the solution of the problem of closed conductors with the EFIE or MFIE impractical.

To correct these issues, the *combined field integral equation (CFIE)* was developed. It can be viewed as a linear combination of the EFIE and MFIE, and is

$$\alpha\left[\frac{1}{2}\overline{\mathbf{J}} + \hat{n}\times\tilde{\mathcal{K}}\{\,\overline{\mathbf{J}}\,\}\right] - \hat{n}\times\hat{n}\times\mathcal{L}\{\,\overline{\mathbf{J}}\,\} = \alpha\hat{n}\times\overline{\mathbf{H}}^{inc}(\mathbf{r}) - \hat{n}\times\hat{n}\times\mathbf{E}^{inc}(\mathbf{r}), \tag{4.115}$$

where α is a positive, real-valued constant that sets the ratio between the EFIE and MFIE for the linear combination. The reason this solution approach works is because it corresponds to an interior problem of a cavity with a wall made from an *impedance boundary condition.* By choosing α to be a positive, real-valued constant we are setting the impedance of the cavity walls to be purely resistive. This shifts the resonances off the real axis and onto the complex plane so that the null space is mathematically eliminated at real frequencies where our analysis is performed.

4.8 Solving the EFIE, MFIE, and CFIE

Having formulated our integral equations in 3D, we now want to consider how to solve each of these equations using the MoM. We will begin by considering the EFIE before considering the MFIE. Due to our discretization approach, the implementation of the CFIE then becomes a trivial combination of the prior two approaches, and so, will not be considered in detail.

4.8.1 Solving the EFIE

To begin the MoM approach, we will need to select a basis and testing function to use. As discussed previously, we will use the RWG function [31]. Recall that each RWG function can be associated with each *interior edge* of a triangular surface mesh, with the function spanning the two triangles that share the interior edge as shown in Fig. 4.9. For the EFIE, it is suitable to use the RWG function as both basis and testing function. Hence, we can expand the current density as

$$\bar{\mathbf{J}}(\mathbf{r}) = \sum_{n=1}^{N} I_n \mathbf{f}_n(\mathbf{r}), \tag{4.116}$$

where N is the total number of interior edges. We can substitute this into the EFIE in (4.113). Prior to testing the equation, we can recognize that since we will be testing with an RWG function the $\hat{n} \times \cdot$ operation in (4.113) to project the equation into the tangential component of the electric field is no longer needed (testing with the RWG function will project the equation onto the tangential components of the surface). Alternatively, we can think of using $\hat{n} \times \mathbf{f}_m$ as the testing function. Either way, when we test (4.113) we end up with a matrix equation of

$$[Z]\{I\} = \{V\}, \tag{4.117}$$

where

$$[Z]_{mn} = \iint_S \mathbf{f}_m(\mathbf{r}) \cdot \mathcal{L}\{\mathbf{f}_n\} dS \tag{4.118}$$

$$\{V\}_m = \iint_S \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}^{inc}(\mathbf{r}) dS. \tag{4.119}$$

A direct numerical implementation of (4.118) can cause difficulties because the gradient operator on $g(\mathbf{r}, \mathbf{r}')$ increases the order of the singularity. A way to overcome this issue is to use integration by parts to transfer this gradient operator onto the testing function. In particular, we get that

$$\begin{aligned}\iint_S \mathbf{f}_m(\mathbf{r}) \cdot \mathcal{L}\{\mathbf{f}_n\} dS &= \iint_S \iint_S \left[jkg(\mathbf{r}, \mathbf{r}') \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') - \mathbf{f}_m(\mathbf{r}) \cdot \nabla g(\mathbf{r}, \mathbf{r}') \frac{\nabla' \cdot \mathbf{f}_n(\mathbf{r}')}{jk} \right] dS' dS \\ &= \iint_S \iint_S \left[jkg(\mathbf{r}, \mathbf{r}') \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') + \frac{g(\mathbf{r}, \mathbf{r}')}{jk} \nabla \cdot \mathbf{f}_m(\mathbf{r}) \nabla' \cdot \mathbf{f}_n(\mathbf{r}') \right] dS' dS.\end{aligned} \tag{4.120}$$

Note that there are no contributions from a boundary integral because it would have an argument like $\hat{n} \cdot [g\, \mathbf{f}_m]$. This is identically zero around the boundary of the surface because all RWG functions have a zero normal component at exterior edges of the mesh by their construction.

Although this operation has reduced the order of the singularity, there still remains a singularity of order $1/R$ for both terms of this double surface integral. A number of different strategies have been developed to accurately evaluate these integrals. One simple strategy, known as *Duffy's method*, is purely numerical in nature and provides a convenient way to integrate the singularity when $\mathbf{f}_m$ and $\mathbf{f}_n$ overlap.

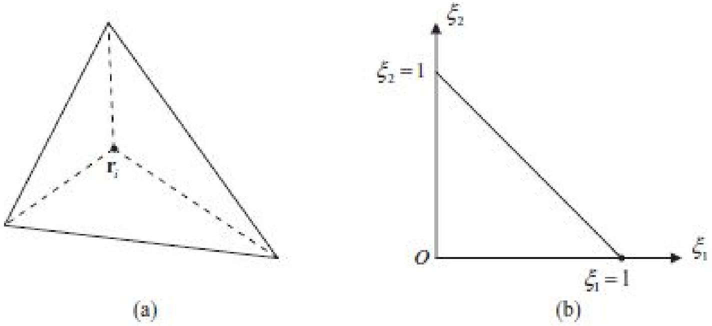


Figure 4.10: (a) Duffy's method uses a quadrature point from the outer quadrature rule to subdivide the original triangle into three subtriangles and (b) each subtriangle is then mapped to a standard right-angled triangle for a further quadrature rule to be applied (images from [5]).

Duffy's method begins by expanding the outer surface integral (i.e., the testing integral) using numerical integration like Gaussian quadrature. For *each* point of this outer quadrature rule, we can write the remaining integral as

$$I = \iint_{\Delta} \frac{f(\mathbf{r}_i, \mathbf{r}')}{|\mathbf{r}_i - \mathbf{r}'|} dS' \quad \mathbf{r}_i \in \Delta, \tag{4.121}$$

where Δ is the triangle being integrated over. Before applying a quadrature rule to the integral given in (4.121), we can first subdivide Δ into three subtriangles by connecting $\mathbf{r}_i$ to each of the vertices of the original triangle, as shown in Fig. 4.10. This allows us to rewrite (4.121) as

$$I = \sum_{e=1}^{3} \iint_{\Delta^e} \frac{f(\mathbf{r}_i, \mathbf{r}')}{|\mathbf{r}_i - \mathbf{r}'|} dS' \quad \mathbf{r}_i \in \Delta. \tag{4.122}$$

Finally, we can evaluate the integrations over each subtriangle by first mapping them to a standard right-angled triangle and applying Gauss-Legendre quadrature on the right-angled triangle. This avoids needing to sample the integral at the singular point because Gauss-Legendre quadrature does not require sampling points on the edge of the integration domain and the singular point of the original integral has been mapped to a vertex of the right-angled triangle. Further, the Jacobian involved in the mapping from the subtriangle to the standard right-angled triangle also regularizes the integral and removes its singularity.

As a result, this method provides a simple and convenient way to handle the singular integrals involved in evaluating the matrix representation of the EFIE. However, requiring the numerical quadrature over three subtriangles *for every point* of the outer integral can increase the computation time. Additionally, high quadrature orders can be needed when the integral is singular or near singular (i.e., two triangles that do not overlap but are still near each other) to achieve a desired accuracy level.

An alternative approach to Duffy's method that can be more efficient is *singularity extraction* or *singularity subtraction* [33]. In these methods, a part of the singular integral that can be evaluated analytically is separated from the overall integration. For example, we can recognize that the Green's function can be rewritten as

$$\frac{e^{jkR}}{4\pi R} = \left(\frac{e^{jkR}-1}{4\pi R}\right) + \frac{1}{4\pi R}. \tag{4.123}$$

If we take the limit as $R \to 0$ of the terms inside the parentheses, we find that this expression is no longer singular. We cannot evaluate this integral in closed form when it is multiplied by RWG functions or the divergence of the RWG functions. However, because it is no longer singular, we can very quickly evaluate it using numerical quadrature rules. The remaining term in (4.123) *can* be integrated analytically when multiplied by an RWG function or the divergence of the RWG function [33]. This provides a very accurate and efficient means to handling this difficult problem. This method can also be applied to near-singular integrals to reduce the number of quadrature points needed in evaluating the terms inside the parentheses in (4.123), improving the efficiency of the method for a desired level of accuracy.

4.8.2 Solving the MFIE

We can also solve the MFIE given in (4.114) using the MoM. We begin in a way similar to the EFIE by expanding the current density in terms of RWG functions. Similarly, we can test the MFIE with an RWG function. This gives us as a matrix equation

$$([G] + [K])\{I\} = \{V\}, \tag{4.124}$$

where

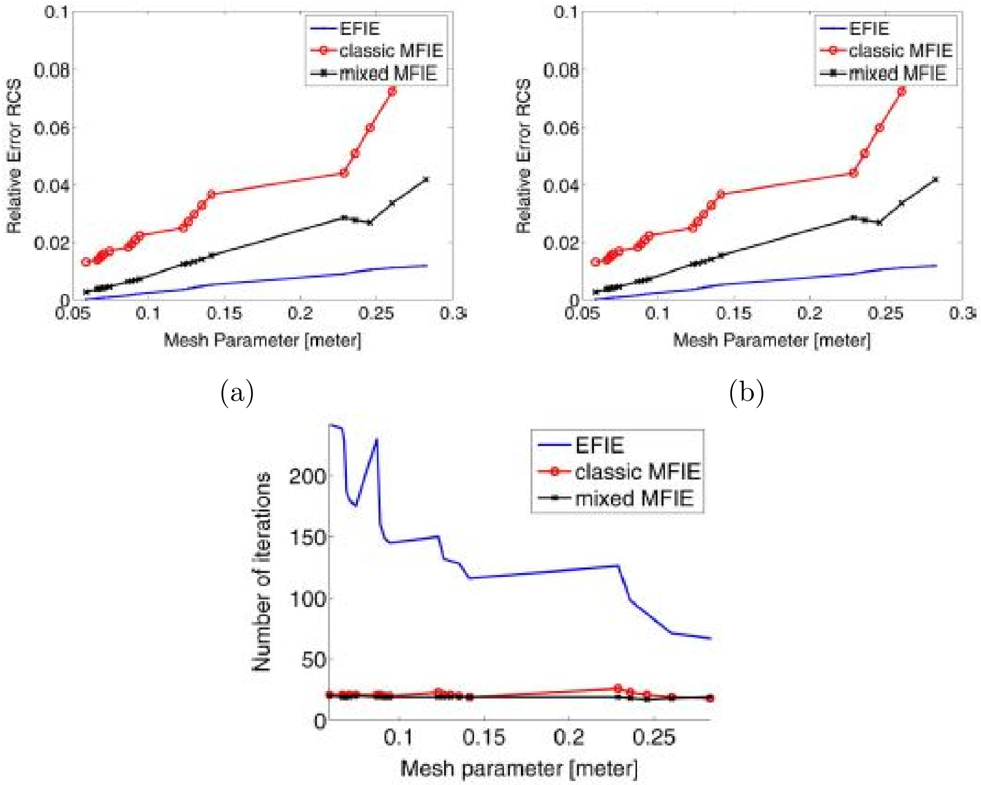
$$[G]_{mn} = \frac{1}{2}\iint_S \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}) dS, \tag{4.125}$$

$$[K]_{mn} = \iint_S \mathbf{f}_m(\mathbf{r}) \cdot \hat{n} \times \tilde{\mathcal{K}}\{\mathbf{f}_n\} dS, \tag{4.126}$$

$$\{V\}_m = \iint_S \mathbf{f}_m(\mathbf{r}) \cdot \hat{n} \times \overline{\mathbf{H}}^{inc}(\mathbf{r}) dS. \tag{4.127}$$

This discretization leads to a well-conditioned integral equation that can typically be solved quickly with iterative solvers. The reason for the well-conditioning is because this matrix equation can be viewed as a matrix representation of the identity operator and a small perturbation. Here, we often refer to the "identity operator" portion as being the *Gram matrix* given in (4.125) (this is a tested form of the identity operator) while the perturbation is given in (4.126). The properties of the RWG functions lead to a well-conditioned Gram matrix, which dominates the overall conditioning of the problem.

Unfortunately, the matrix representation of the "perturbation" given in (4.126) is not tested well when we use an RWG function as both basis and testing function [16]. This leads to the MFIE having a lower accuracy than the EFIE. This problem can be almost



(a)

(b)

(c)

Figure 4.11: Analysis of scattering from a 1 meter cube using different integral equation formulations. “Classic MFIE” corresponds to Galerkin testing, while “mixed MFIE” corresponds to a discretization that conforms to the Sobolev space properties of the MFIE. (a) Relative error as a function of “average” mesh length, (b) relative error as a function of frequency, and (c) convergence history (images from [16]).

completely resolved by instead testing the MFIE with a different function, known as a *Buffa-Christensen function* [32]. This represents an important case where Galerkin’s method can be proven to lead to poorer results than other discretization approaches [16]. Numerical results demonstrating this are included in Fig. 4.11.

Just as with the EFIE, the MFIE operator has a singularity to it that must be handled carefully to ensure accurate numerical results. To address this, similar strategies as were used for the EFIE can be applied to the MFIE [33]. One important point to remember about the MFIE is that it is to be evaluated in a principal value sense. As a result, the integrals contained in $[K]$ should not go over the exactly singular point.

4.8.3 CFIE Example

As mentioned previously, the CFIE matrix can be formulated as a trivial extension of the methods discussed for the EFIE and MFIE. Here, we only show the results of the CFIE for analyzing the induced surface current density on a PEC almond shape. This is a standard

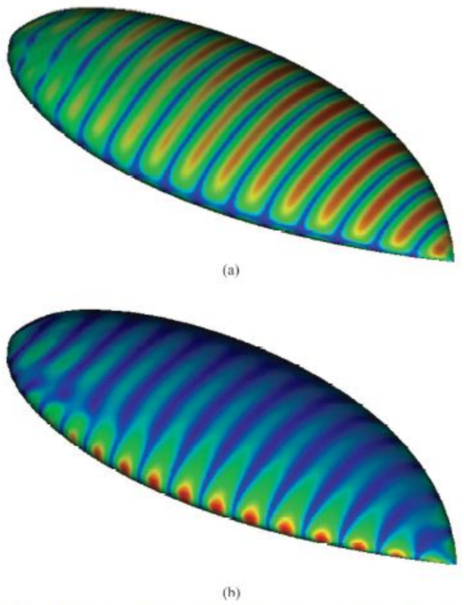


Figure 10.19 Snapshot of the induced surface electric current density on a 9.936-inch-long conducting almond with a 10-GHz plane wave incident horizontally from an azimuth angle of 30° away from the tip. (a) Vertical polarization. (b) Horizontal polarization.

Figure 4.12: Surface current density on the almond shape (images from [5]).

test object for CEM codes, with historical measurement data available to use for validation purposes [5]. Due to the difficulty in accurately measuring the radar cross section of objects, this validation is predominantly useful for checking general trends in the data since minor misalignments of the object can significantly perturb the results at a single frequency comparison. The data from this analysis is shown in Figs. 4.12 and 4.13.

4.9 Analyzing Penetrable Media

Up to now, we have only considered the case of an impenetrable 3D object; namely, a PEC scatterer. We must update our integral equation formulation when we want to consider a penetrable region, such as a dielectric or magnetic scatterer. We will now briefly consider a few options for this situation.

To begin, recall that before we used the fact that $\mathbf{M} = -\hat{n} \times \mathbf{E} = 0$ on a PEC surface, the integral equations that we formulated for our 3D exterior equivalent problem were

$$\frac{1}{2}\mathbf{M} + \hat{n} \times \tilde{\mathcal{K}}_e\{\mathbf{M}\} - \hat{n} \times \mathcal{L}_e\{\,\overline{\mathbf{J}}\,\} = -\hat{n} \times \mathbf{E}^{inc}(\mathbf{r}) \quad \mathbf{r} \in S_-, \tag{4.128}$$

$$\frac{1}{2}\overline{\mathbf{J}} + \hat{n} \times \tilde{\mathcal{K}}_e\{\,\overline{\mathbf{J}}\,\} + \hat{n} \times \mathcal{L}_e\{\mathbf{M}\} = \hat{n} \times \overline{\mathbf{H}}^{inc}(\mathbf{r}) \quad \mathbf{r} \in S_-. \tag{4.129}$$

Note that the subscripts of e on the different integral operators are to remind us that these were formulated for an *exterior problem*, and hence the material properties that should be

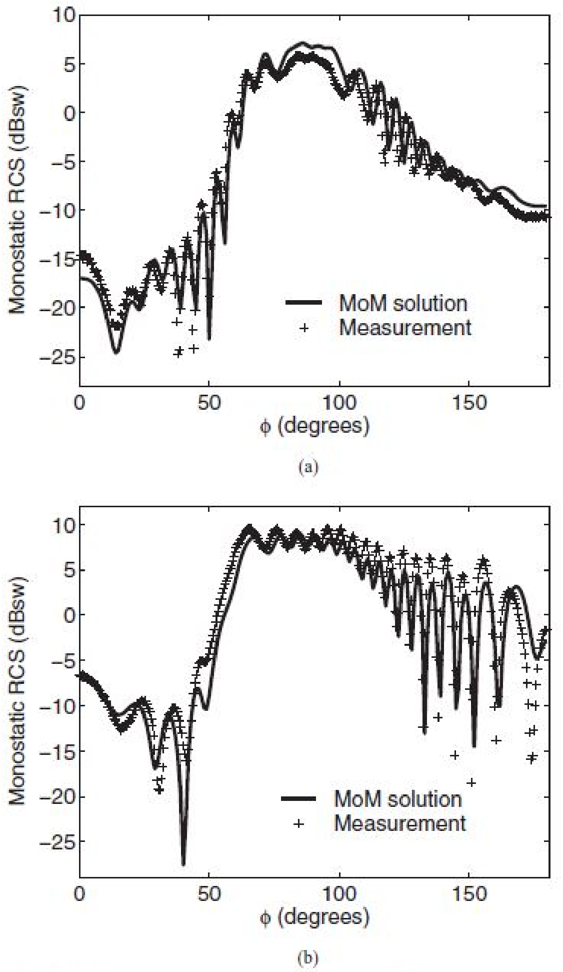


Figure 10.20 Monostatic RCS of a 9.936-inch-long conducting almond at 10 GHz. (a) VV polarization. (b) HH polarization.

Figure 4.13: Monostatic radar cross section for the almond shape (images from [5]).

used in evaluating the different equations are those from the exterior region. We also see that since there are two unknowns in each equation, we cannot implement an EFIE or MFIE in an identical manner to how we did for PEC objects.

The way to solve this issue is to formulate additional integral equations from the perspective on an *interior equivalent problem*. This can be formulated using a similar surface equivalence principle approach to what we did previously for the exterior problem. One change is that because we are coming from the interior region there is no longer an incident field (assuming there are no current sources inside the dielectric object we are analyzing). Another change is that we must modify how we go about deforming our integration domain to evaluate the principal parts of the different integral operators. This leads to a change in sign of the resulting component. Combining these changes, the integral equations for the interior problem become

$$-\frac{1}{2}\mathbf{M} + \hat{n} \times \tilde{\mathcal{K}}_i\{\mathbf{M}\} - \hat{n} \times \mathcal{L}_i\{\, \eta_i \overline{\mathbf{J}}\,\} = 0 \quad \mathbf{r} \in S_+, \tag{4.130}$$

$$-\frac{1}{2}\eta_i\overline{\mathbf{J}} + \hat{n}\times\tilde{\mathcal{K}}_i\{\,\eta_i\overline{\mathbf{J}}\,\} + \hat{n}\times\mathcal{L}_i\{\mathbf{M}\} = 0 \quad \mathbf{r}\in S_+, \tag{4.131}$$

where $\eta_i = \sqrt{\mu_i\epsilon_e/\mu_e\epsilon_i}$. This scaling factor is necessary to correct for the scaling factors embedded in the definition of $\overline{\mathbf{J}}$ and the different integral operators.

The EFIE for a penetrable region then consists of solving (4.128) and (4.130) together. Likewise, the MFIE for a penetrable region consists of solving (4.129) and (4.131) together. Although these are valid options, both of these integral equations suffer from interior resonances. There also exist some complications around achieving a well-performing discretization of these equations if only RWG functions are used.

One approach that avoids both of these issues is to use all four integral equations that have been formulated up to this point. This approach is known as the *PMCHWT formulation*, and is named after all the different authors that significantly contributed to its original formulation in a sequence of papers. In this formulation, the EFIE for the exterior and interior equivalent problems are added together to form a single equation, with a similar process done for the two MFIEs. The resulting equation system is given as

$$\hat{n}\times\left[\mathcal{L}_e\{\,\overline{\mathbf{J}}\,\} + \mathcal{L}_i\{\,\eta_i\overline{\mathbf{J}}\,\}\right] - \hat{n}\times\left[\tilde{\mathcal{K}}_e\{\mathbf{M}\} + \tilde{\mathcal{K}}_i\{\mathbf{M}\}\right] = \hat{n}\times\mathbf{E}^{inc}(\mathbf{r}) \quad \mathbf{r}\in S, \tag{4.132}$$

$$\hat{n}\times\left[\tilde{\mathcal{K}}_e\{\,\overline{\mathbf{J}}\,\} + \tilde{\mathcal{K}}_i\{\,\overline{\mathbf{J}}\,\}\right] + \hat{n}\times\left[\mathcal{L}_e\{\mathbf{M}\,\} + \mathcal{L}_i\{\eta_i^{-1}\mathbf{M}\,\}\right] = \hat{n}\times\overline{\mathbf{H}}^{inc}(\mathbf{r}) \quad \mathbf{r}\in S. \tag{4.133}$$

Physically, we can recognize that the operators on the left-hand sides of both equations compute the scattered electric and magnetic fields in the interior and exterior regions. Considering this, we see that (4.132) corresponds to enforcing that the tangential components of the electric field is continuous at the interface of the penetrable region, with a similar interpretation for the magnetic field in (4.133). As alluded to previously, this set of equations is free from interior resonances and RWG functions can be used as both basis and testing functions for all quantities involved. Although this is useful, the particular combination of exterior and interior integral equations has removed the "identity operators" so that this is an integral equation of the first kind. As a result, it tends to be ill-conditioned and can be difficult to solve numerically with iterative solvers.

Another alternative integral equation for penetrable regions is the *Müller formulation.* Essentially, this formulation subtracts the two EFIEs and two MFIEs from one another as opposed to adding them. This leads to a second kind integral equation that can lead to a significantly better conditioned matrix system than the PMCHWT. However, to achieve the best performance it is necessary to use both RWG and Buffa-Christensen functions in the discretization process, making the numerical method more complex to implement.

4.10 Introduction to Fast Algorithms

One of the major differences between the matrix systems generated from finite difference or finite element methods and the method of moments was that the differential equation solvers yielded extremely sparse matrices, while integral equation solvers gave a completely dense matrix due to the presence of the Green's function. This has incredibly important consequences on the utility of these methods to problems with large numbers of unknowns. We discussed at various times in class that by exploiting the sparsity of the FDM or FEM matrices we could keep the computational and memory complexity of the methods to reasonably manageable levels.

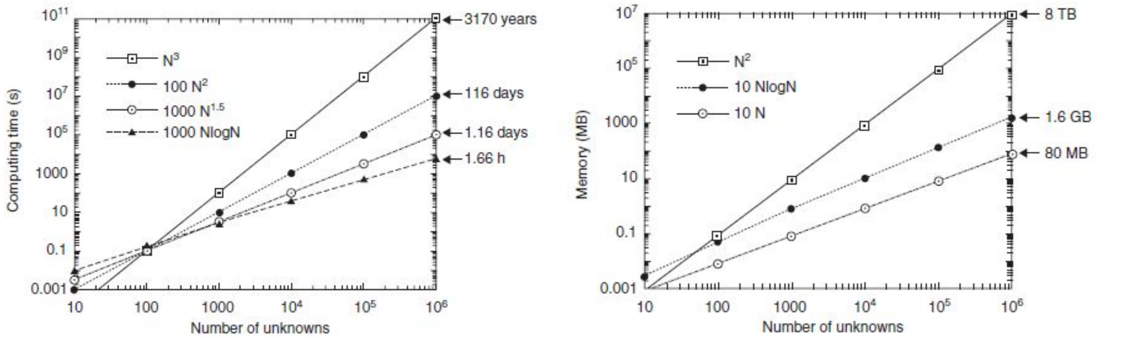
This story is no longer the same for the MoM, since the full matrix results in a memory complexity of $O(N^2)$ and the computational complexity of directly solving a system like this is of $O(N^3)$. This quickly becomes impossible to solve for large problems that are commonly encountered in engineering design, as illustrated in Fig. 4.14. We can of course use an iterative solver instead of a direct solver to potentially improve the computational complexity of our methods. However, because we are still dealing with a full matrix, each matrix-vector product requires $O(N^2)$ operations. Hence, the iterative solver complexity will be on $O(N_{\text{iter}}N^2)$ where N_{iter} is the number of iterations required to reach a desired convergence level. Considering Fig. 4.14, we see that although this does extend the range of problems that can potentially be solved, it still falls far short of what would be needed to tackle realistic engineering problems.

For many years, these computational bottlenecks appeared to severely limit the long-term feasibility of using the MoM for practical engineering analyzes, even accounting for them only requiring a surface discretization of the geometries of interest. These issues were eventually addressed through the creation of *fast algorithms*. In the context of CEM, fast algorithms almost always refers to a type of method used to speed up the solution of an integral equation solver. Originally, these methods were only applicable to speeding up the evaluation of matrix-vector products, and hence, were only relevant to the acceleration of iterative solvers. However, more recently, new classes of fast algorithms have begun to be developed that can be applied to either iterative or direct solvers. With the advent of these techniques, the computational complexity of integral equation solvers was able to reach $O(N \log N)$, greatly increasing the applicability and popularity of integral equation solvers. This has made them become the standard approach for analyzing electrically large problems when the accuracy of a full-wave solution is needed. Although this is still generally the case, it should be noted that advanced techniques applied to differential equation solvers can still allow them to be applied to very large scale problems as well. As a result, all of these methods continue to be areas of active research interest.

4.11 Fast Multipole Method – 2D Case

The fast multipole method (FMM) is an early fast algorithm that is designed to accelerate the computation of matrix-vector products with MoM matrices. This acceleration can greatly increase the efficiency of iterative solvers in the solution of electromagnetic integral equations. We will review this method in the simpler 2D case to get the general idea, and then eventually discuss how it can be extended into a multilevel algorithm known as the multilevel fast multipole algorithm (MLFMA). It is the MLFMA that achieves the much sought after computational complexity of $O(N \log N)$, which makes it one of the most successful fast algorithms to date.

Now, our goal with the FMM method is to accelerate the evaluation of a matrix-vector



(a) (b)

Figure 4.14: Illustration of the need for fast algorithms through (a) computational complexity and (b) memory complexity for a few hypothetical numerical schemes (images from [5]). Computation time estimates are based on the performance of single-core processors from the 2010's.

product with the fully dense MoM matrix. To see the basic process of the FMM, we will consider the TM polarization case of scattering from a conducting cylinder (although the FMM can be applied to both polarizations and penetrable scatterers as well). For this situation, the MoM matrix was given by

$$[Z]_{mn} = \frac{k\eta}{4} \int_S t_m(\boldsymbol{\rho}) \int_S H_0^{(2)}(k|\boldsymbol{\rho} - \boldsymbol{\rho}'|) f_n(\boldsymbol{\rho}') dS' dS, \tag{4.134}$$

where t_m and f_n denote the testing and basis functions, respectively.

From a physical perspective, we can interpret each matrix element of $[Z]$ as being the field radiated by a current element with shape f_n that is received by another current element with shape t_m. Considering this, the inner product of any row of $[Z]$ with the coefficient vector of the basis functions $\{J\}$ can be interpreted as the total field radiated by all current elements on the scatterer that is received by t_m. Doing this explicitly for a single current element takes $O(N)$ operations, and repeating this process for all current elements naturally extends this to the full $O(N^2)$ operations of the matrix-vector product.

The FMM process speeds this computation up by recognizing that when one is "far" away from a set of current sources, the observer is only able to distinguish the combined fields and is not able to distinguish the effect of every single current source that produced the combined field. With enough distance, the combined field can be represented using a much smaller number of parameters compared to considering the effect of every source individually. For instance, when an observer is far away from a complicated antenna array, all that the observer sees is a simple plane wave. This plane wave may be described in a much simpler manner than considering the potentially thousands of individual array elements that produced it.

To actually implement this mathematically for practically relevant scenarios, the FMM must make use of some clever mathematical manipulations of the Green's function in (4.134). To facilitate this, part of the "initialization" of the FMM process is to first subdivide all of the basis functions on the scatterer into groups of elements based on their spatial proximity

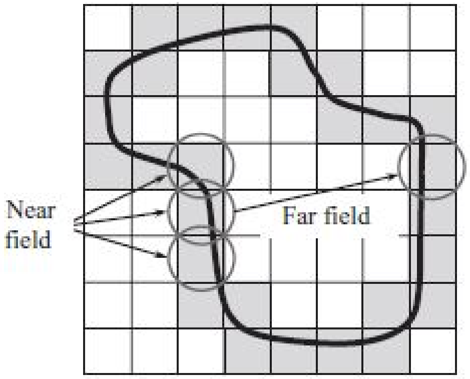


Figure 4.15: Illustration of near field and far field groups for the FMM (image from [5]).

to one another, as shown in Fig. 4.15. For each group, we can then decide whether all other groups should be classified as *near field* or *far field* groups. Near field groups are ones that are located in a close proximity to one another such that it does not make physical sense to try and "compress" the interactions between them. For these groups, the MoM matrix is explicitly calculated and stored for direct use in the evaluation of matrix-vector products. In contrast to this, far field groups have enough physical distance between them that the interaction between current elements in the two groups can be more efficiently evaluated.

The particular strategy for efficiently evaluating the far field interactions is to break the computation up into three processes. The first step, known as *aggregation*, involves "lumping" together the radiation of all basis functions within a group so that it can be described as if it is radiating from the center of the group. Next, the effect of the aggregated fields is *translated* from the source group center to the center of the observer's group. This effect is then *disaggregated* by determining how the radiation received at this group center should effect all of the individual current elements in this group. Mathematically, these effects are facilitated through use of the *addition theorem*.

To assist in this process, we will first introduce some notation for the different groups. We will denote the group to which a source function $f_n(\boldsymbol{\rho}')$ belongs as G_q, whose center lies at $\boldsymbol{\rho}_q$. The group to which a testing function $t_m(\boldsymbol{\rho})$ belongs will be denoted as G_p, whose center lies at $\boldsymbol{\rho}_p$. We will now need to factorize the Hankel function into three pieces to represent the different stages of the FMM algorithm. The first piece will need to depend on $\boldsymbol{\rho}_q - \boldsymbol{\rho}'$ to achieve the aggregation, the second piece will need to depend on $\boldsymbol{\rho}_{pq} = \boldsymbol{\rho}_p - \boldsymbol{\rho}_q$ to achieve the translation, and finally the third piece will need to depend on $\boldsymbol{\rho} - \boldsymbol{\rho}_p$ to achieve the disaggregation. Considering this, we can write $\boldsymbol{\rho} - \boldsymbol{\rho}'$ as

$$\boldsymbol{\rho} - \boldsymbol{\rho}' = (\boldsymbol{\rho} - \boldsymbol{\rho}_p) + \boldsymbol{\rho}_{pq} + (\boldsymbol{\rho}_q - \boldsymbol{\rho}'), \tag{4.135}$$

which is illustrated in Fig. 4.16.

We now turn to using the decomposition of the vector $\boldsymbol{\rho} - \boldsymbol{\rho}'$ shown in (4.135) in the addition theorem for the Hankel function. Recall that the addition theorem provides a way to expand an "off-centered" Hankel function in terms of a summation of Hankel functions

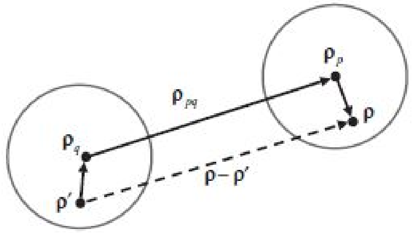


Figure 4.16: Illustration of decomposition of $\boldsymbol{\rho}-\boldsymbol{\rho}'$ into three pieces for use in the FMM (image from [5]).

"centered" at a different location as

$$H_0^{(2)}(k|\boldsymbol{\rho}_c+\boldsymbol{\rho}_d|) = \sum_{l=-\infty}^{\infty} J_l(k\rho_d) H_l^{(2)}(k\rho_c) e^{jl(\phi-\phi_d-\pi)}, \quad \rho_c > \rho_d, \tag{4.136}$$

where $\boldsymbol{\rho}_c$ and $\boldsymbol{\rho}_d$ are radial position vectors in a cylindrical coordinate system and ϕ_d is the angle between $\boldsymbol{\rho}_d$ and the x-axis. Before specifying how we will relate $\boldsymbol{\rho}_c$ and $\boldsymbol{\rho}_d$ to (4.135), it is useful to rewrite (4.136) using the integral representation of the Bessel function

$$J_l(k\rho_d)e^{-jl(\phi_d+\pi)} = \frac{1}{2\pi}\int_0^{2\pi} e^{-j\mathbf{k}\cdot\boldsymbol{\rho}_d - jl(\alpha+\pi/2)}\, d\alpha, \tag{4.137}$$

where $\mathbf{k} = k(\hat{x}\cos\alpha + \hat{y}\sin\alpha)$. From an intuitive perspective, we can view (4.137) as being a plane-wave expansion of a cylindrical wave. This "change of basis" will be key in building the collective "radiation" and "receive" patterns of the various groups of current elements. To see this, we substitute (4.137) into (4.136) to get

$$H_0^{(2)}(k|\boldsymbol{\rho}_c+\boldsymbol{\rho}_d|) = \frac{1}{2\pi}\sum_{l=-\infty}^{\infty} H_l^{(2)}(k\rho_c) e^{jl\phi} \int_0^{2\pi} e^{-j\mathbf{k}\cdot\boldsymbol{\rho}_d - jl(\alpha+\pi/2)}\, d\alpha, \quad \rho_c > \rho_d. \tag{4.138}$$

By now setting $\boldsymbol{\rho}_c = \boldsymbol{\rho}_{pq}$ and $\boldsymbol{\rho}_d = (\boldsymbol{\rho}-\boldsymbol{\rho}_p) + (\boldsymbol{\rho}_q - \boldsymbol{\rho}')$ in (4.138), we get

$$H_0^{(2)}(k|\boldsymbol{\rho}-\boldsymbol{\rho}'|) = \frac{1}{2\pi}\int_0^{2\pi} e^{-j\mathbf{k}\cdot(\boldsymbol{\rho}-\boldsymbol{\rho}_p)}\tilde{\alpha}_{pq}(\alpha) e^{-j\mathbf{k}\cdot(\boldsymbol{\rho}_q-\boldsymbol{\rho}')}, \quad \rho_{pq} > |(\boldsymbol{\rho}-\boldsymbol{\rho}_p) + (\boldsymbol{\rho}_q - \boldsymbol{\rho}')| \tag{4.139}$$

where

$$\tilde{\alpha}_{pq}(\alpha) \approx \sum_{l=-L}^{L} H_l^{(2)}(k\rho_{pq}) e^{jl(\phi_{pq}-\alpha-\pi/2)}. \tag{4.140}$$

In (4.140), ϕ_{pq} is the angle that $\boldsymbol{\rho}_{pq}$ makes with the x-axis and the infinite series of Hankel functions has been truncated to make the identity useful for practical computations. Importantly, error bounds can be derived to determine what L should be to achieve a desired level of accuracy, making this representation of the Green's function *error controllable*.

We may now use (4.139) to speed up the computation of far-field interactions. So long as the groups G_p and G_q are not neighbors (this is necessary to satisfy the condition that $\rho_{pq} > |(\boldsymbol{\rho} - \boldsymbol{\rho}_p) + (\boldsymbol{\rho}_q - \boldsymbol{\rho}')|$), the interaction between two basis and testing functions given by (4.134) can be written as

$$[Z]_{mn} = \frac{k\eta}{8\pi} \int_0^{2\pi} \left[\int_S t_m(\boldsymbol{\rho}) e^{-j\mathbf{k}\cdot(\boldsymbol{\rho}-\boldsymbol{\rho}_p)} dS \right] \tilde{\alpha}_{pq}(\alpha) \left[\int_S f_n(\boldsymbol{\rho}') e^{-j\mathbf{k}\cdot(\boldsymbol{\rho}_q-\boldsymbol{\rho}')} dS' \right] d\alpha. \tag{4.141}$$

To keep the notation more concise, we can define the receive functions and radiation functions as

$$\tilde{t}_{mp}(\alpha) = \int_S t_m(\boldsymbol{\rho}) e^{-j\mathbf{k}\cdot(\boldsymbol{\rho}-\boldsymbol{\rho}_p)} dS \tag{4.142}$$

and

$$\tilde{f}_{nq}(\alpha) = \int_S f_n(\boldsymbol{\rho}') e^{-j\mathbf{k}\cdot(\boldsymbol{\rho}_q-\boldsymbol{\rho}')} dS', \tag{4.143}$$

respectively, so that (4.141) becomes

$$[Z]_{mn} = \frac{k\eta}{8\pi} \int_0^{2\pi} \tilde{t}_{mp}(\alpha) \tilde{\alpha}_{pq}(\alpha) \tilde{f}_{nq}(\alpha) d\alpha. \tag{4.144}$$

Now, the matrix-vector product between the MoM impedance matrix and the basis function expansion coefficients for a particular row of the matrix can be written as

$$\begin{aligned} \sum_{n=1}^{N} [Z]_{mn} \{J_z\}_n = &\sum_{q\in B_p} \sum_{n\in G_q} [Z]_{mn} \{J_z\}_n \\ &+ \frac{k\eta}{8\pi} \int_0^{2\pi} \tilde{t}_{mp}(\alpha) \sum_{q\notin B_p} \tilde{\alpha}_{pq}(\alpha) \sum_{n\in G_q} \tilde{f}_{nq}(\alpha) \{J_z\}_n d\alpha, \qquad m \in G_p \end{aligned} \tag{4.145}$$

where B_p denotes the set of groups directly neighboring G_p and G_p itself. Considering this, the first term in (4.145) represents the near-field interactions that are computed directly, while the second term is the contribution of all far-field interactions. For practical computations, the integral over α can be approximated using a numerical quadrature rule so that it is replaced by a sum over R points, where R is proportional to the number of current elements in the group. This allows us to finally write the interactions as

$$\begin{aligned} \sum_{n=1}^{N} [Z]_{mn} \{J_z\}_n = &\sum_{q\in B_p} \sum_{n\in G_q} [Z]_{mn} \{J_z\}_n \\ &+ \frac{k\eta}{4R} \sum_{r=1}^{R} \tilde{t}_{mp}(\alpha_r) \sum_{q\notin B_p} \tilde{\alpha}_{pq}(\alpha_r) \sum_{n\in G_q} \tilde{f}_{nq}(\alpha_r) \{J_z\}_n, \qquad m \in G_p. \end{aligned} \tag{4.146}$$

To see how this factorization speeds up the computation, it is useful to count the number of operations needed in evaluating (4.146). To begin, we will assume that we have subdivided the basis functions such that each group contains approximately M basis functions. For this case, we will have that the first term in (4.146) that represents the near-field interactions can be evaluated in $C_g M^2 \times N/M = C_g MN$ operations, where C_g is the number of near-field groups (in 2D, this is typically $C_g = 3$). This comes from the M^2 interactions that need to be evaluated for every near-field group times the total number of groups, which is proportional to N/M.

To count the operations for the far-field interactions, it is easiest to count the operations for each stage of the algorithm. The aggregation step requires the calculation of the sum

$$F_{qr} = \sum_{n \in G_q} \tilde{f}_{qn}(\alpha_r)\{J_z\}_n, \quad q = 1, 2, \ldots, N/M; \; r = 1, 2, \ldots, R, \tag{4.147}$$

which can be completed in $R \times M \times N/M \approx NM$ operations by recalling that $R \approx M$. As mentioned previously, this step of the algorithm involves lumping the fields radiated by all sources in group G_q to its group center. The next step is to translate the fields from all far-field group centers to the center of group G_p. This is accomplished by

$$F_{pr} = \sum_{q \notin B_p} \tilde{\alpha}_{pq}(\alpha_r) F_{qr}, \quad q = 1, 2, \ldots, N/M; \; r = 1, 2, \ldots, R, \tag{4.148}$$

which takes $R \times (N/M)^2 \approx N^2/M$ operations. The final step of disaggregation requires the calculation of the sum

$$F_{mp} = \sum_{r=1}^{R} \tilde{t}_{mp}(\alpha_r) F_{pr}, \quad m = 1, 2, \ldots, N, \tag{4.149}$$

which can be completed in $R \times N \approx NM$ operations. As mentioned previously, this step distributes the fields received at the group center to each testing function within G_p.

Adding all of these steps up *for all* m, the total computation time of the matrix-vector product $[Z]\{J_z\}$ is

$$T_{\text{op}} = C_1 NM + C_2 N^2/M, \tag{4.150}$$

where C_1 and C_2 are constants. This total computation time reaches its minimum of $T_{\text{min,op}} = 2\sqrt{C_1 C_2} N^{3/2}$ when $M = \sqrt{C_2 N/C_1} \approx \sqrt{N}$. Considering this, we see that the total computational complexity has been reduced from $O(N^2)$ to $O(N^{3/2})$. Similarly, the memory required also has reduced from $O(N^2)$ to $O(N^{3/2})$.

From Fig. 4.14, we see that this is a good increase in performance when N is large, but that this is still not sufficient for many large-scale engineering problems. To address this, we need to develop a multilevel version of FMM that can further reduce the computational and memory complexities of the algorithm. We will discuss this algorithm at a high-level in the coming lecture.

4.12 Overview of the Multilevel Fast Multipole Algorithm (MLFMA)

Previously, we saw that we could accelerate the evaluation of a matrix-vector product with a MoM matrix by using the FMM. This involved factorizing the Green's function into three parts so that we could aggregate the effects of a group of nearby basis functions to a single point at the group center, translate the effects of the radiation of this aggregated source to a central point at another far away group, and then disaggregate the results to each basis function in the receiving group. If we were clever with how many basis functions we kept in each group, we saw that we were able to reduce the computational complexity of the matrix-vector product from $O(N^2)$ to $O(N^{3/2})$. Although this was good progress, it is not sufficient to handle the full scope of engineering problems that need to be solved. To address this, the multilevel fast multipole algorithm (MLFMA) was developed.

The basic idea of the MLFMA is to circumvent the key tradeoff in the efficiency of the FMM. In particular, for the FMM with N unknowns divided into N/M groups (where M is the approximate number of basis functions in each group), the calculation of all near-field interactions and the aggregation and disaggregation steps required $O(NM)$ operations. Our analysis also showed that the translation step required $O(N^2/M)$ operations. Hence, if we make M large, we reduce the translation computations but increase the computation involved in the remaining steps. As a result, $M \approx \sqrt{N}$ to approximately balance the computation time involved in all the different parts of the FMM. The MLFMA circumvents this by recognizing that we can apply the FMM process in multiple levels by aggregating the effects of nearby groups together before performing the translation and corresponding multilevel disaggregation. This allows us to keep the number of basis functions in the lowest level groups low so that near-field interactions scale approximately as $O(N)$, but the translation step will no longer depend on $O(N^2/M)$.

To see why the number of translation steps reduces, it is useful to make an analogy to a telephone network (see Fig. 4.17). A very inefficient telephone network with N phones would go about connecting the phones to each other by making direct connections between every phone in the network, as illustrated in Fig. 4.17(a). This approach is equivalent to the direct MoM where every current element interacts with every other current element directly. A more efficient approach to connecting the different telephone users together would be to use local *hubs*. By then wiring the different hubs together, the total number of telephone lines that have to be utilized is greatly reduced, as illustrated in Fig. 4.17(b). This approach is equivalent to the FMM, which grouped current elements together and then translated the effects of each group to other groups where this computation was acceptable (e.g., "far-field" interactions). Although using a single layer of hubs is beneficial, there are still many telephone lines being used in a large network. To further reduce the number of lines, additional layers of hubs can be used to interconnect the lower level hubs, as shown in Fig. 4.17(c). This is the idea of the MLFMA, which will use groups of different sizes as needed to minimize the number of operations needed to compute the interaction between any given set of current elements in the problem. This eventually allows it to reduce the computational complexity of a matrix-vector product to be $O(N \log N)$.

The standard way to perform the grouping of basis functions in the MLFMA is to begin by enclosing the entire object of interest in a single cube. This cube is then evenly divided into eight smaller cubes. Each subcube can then be further subdivided into eight smaller cubes in a recursive process until the smallest cubes contain the desired number of basis functions. This is typically implemented using an *octree data structure*, which also is commonly used in 3D graphics engines for subdividing 3D space. An example of this subdivision process is shown in 2D in Fig. 4.18.

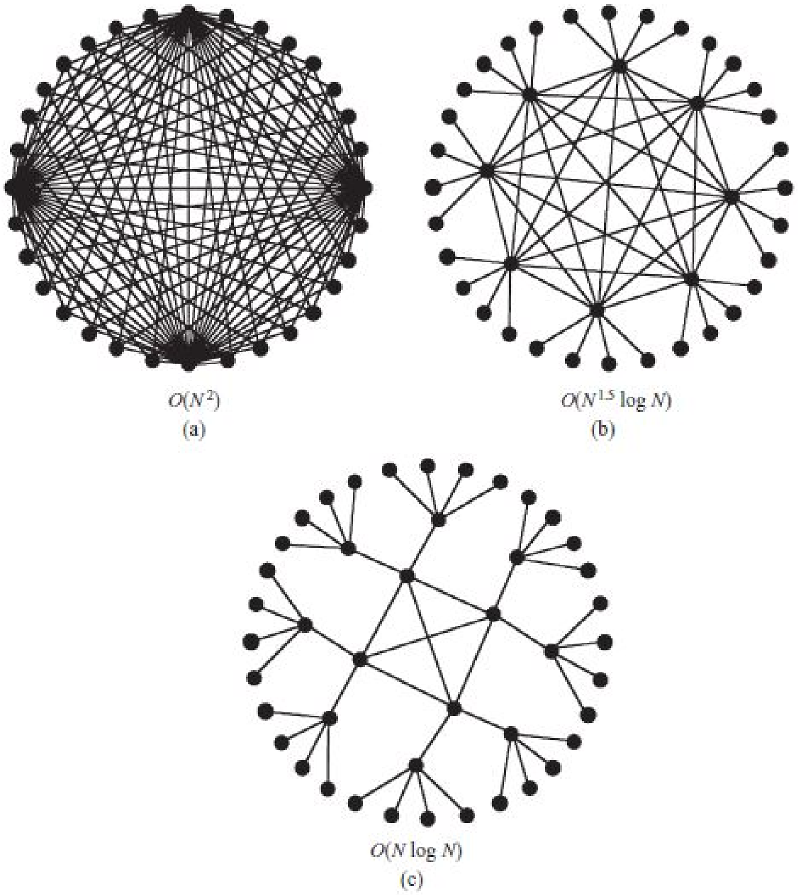


Figure 4.17: Example of a telephone communication network using different connection strategies. (a) Direct connections between every user (only 4 users are actually fully interconnected for illustration purposes), (b) connections via a single layer of hubs, and (c) connections using multiple layers of hubs (images from [5]).

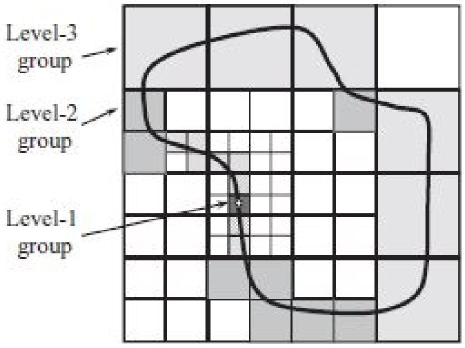


Figure 4.18: Subdivision of a structure into multilevel groups for use in the MLFMA (image from [5]).

Using the MLFMA, unprecedentedly large integral equations became possible to be

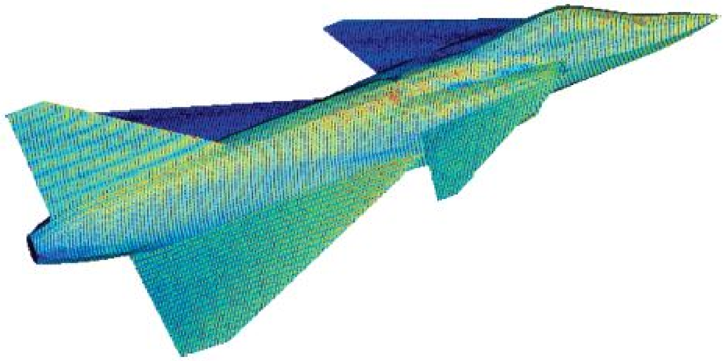


Figure 4.19: Surface currents plotted on an airplane that were solved using the MLFMA (image from [34]).

solved. As an example, the scattering of a plane wave by an airplane at 2 GHz was computed using the MLFMA. At this frequency, the airplane was longer than 100λ long, requiring a surface discretization of nearly one million unknowns. Solving this problem directly with the MoM would have required 8 TB of memory and would have taken a prohibitive amount of computer time (thousands of years). However, with the MLFMA the memory requirement was reduced to 2.5 GB and was able to be solved within a realistic timeframe. As an example of the results produced by this method, the surface currents on the airplane are plotted in Fig. 4.19.

4.13 Adaptive Cross Approximation (ACA)

Previously, we learned about the FMM and MLFMA fast algorithms. These fast algorithms are sometimes referred to as *physics-based* methods because they involved an explicit factorization of the Green's function based on our mathematical and physical insight. Alternatively, these methods are also sometimes referred to as *kernel-based* because they involve explicit operations with the kernel of the integral equation (in this case, the Green's function or derivatives of the Green's functions). Although these methods are very successful, one major drawback is that if we have a new integral kernel we must completely reformulate how we will go about factorizing the new integral kernel and how to numerically implement the factorization efficiently (assuming we even can determine a suitable factorization). Even though the integral kernels we discussed for the EFIE and MFIE can handle very general cases, other electromagnetic integral kernels can be devised to increase the efficiency of our methods for specific applications. These include integral kernels formulated for layered medium problems and other specialized geometries, such as coplanar waveguides. In these situations, the FMM or MLFMA would need to be reformulated for these new integral kernels.

There also exist an alternative class of fast algorithms, known as *algebra-based* or *kernel-independent* methods. These methods work directly on the elements of the MoM matrix and achieve the computational efficiency improvements by using clever numerical linear algebra manipulations. These formulations only require the explicit computation of a small number of elements of the MoM matrix, which can be done easily using a previously developed MoM code. Importantly, because no explicit factorization of the integral kernel is needed, these methods can be easily applied to different integral equation problems very easily. However, the linear algebra techniques used only approximately implement the factorization of the Green's function that was explicitly handled in the FMM or MLFMA methods. As a result, the performance improvements of the algebra-based methods is not always able to match that of the physics-based methods (however, they are still sizable improvements compared to a direct MoM solution). We will only discuss one algebra-based method in this course, the *adaptive cross approximation (ACA)*, but this area remains a very active research area for accelerating electromagnetic integral equations.

4.13.1 Low-Rank Matrix Representations

To understand the working principle of the ACA, it is first necessary to consider some details about how the MoM matrix can be divided into subblocks with different *matrix rank*. First, recall that the rank of a matrix corresponds to the maximum number of linearly independent columns of the matrix. Obviously, the full MoM matrix must be "full rank" (i.e., all columns are linearly independent) for the problem to be invertible. However, in many practical situations, it can occur that even though all columns of a matrix may be linearly independent the "amount" or "degree" of this independence may be small for certain portions of the matrix (e.g., they are close to being linearly dependent).

This can be even more common if we focus on only a small part of an overall matrix. For instance, we can focus on the subblock of a MoM matrix that corresponds to the interactions between sets of current elements that are far apart from each other. As discussed in the context of the FMM, the distance between the sets of currents causes them to only be able to see the "collective effect" of many current elements rather than the individual details due to every current element. This collective effect can be characterized using a small number of parameters, which implies that there may exist a more efficient way to represent this subblock of the MoM matrix than explicitly storing every matrix element. This is indeed the case, and the ACA provides a particular strategy for efficiently computing this reduced representation of the MoM subblock. From a terminology perspective, we would typically refer to these subblocks as being *low-rank* or *rank-deficient*. In contrast to this, subblocks of the MoM matrix that represent "near-field" interactions would not typically be able to be more efficiently stored, and so would be considered *full-rank* matrices.

To see the connection between how the MoM matrix is generated and its potential rank-deficient nature, it is instructive to look at a subblock $[z]_{M\times M}$ with elements defined by

$$[z]_{mn} = \int_S \int_S \psi_m(\mathbf{r}) g(\mathbf{r},\mathbf{r}') \psi_n(\mathbf{r}') dS' dS, \tag{4.151}$$

where $M \ll N$ and is assumed to be the number of basis and testing functions in two far apart groups. Note that we assume scalar functions here to keep the notation simpler, but the basic conclusions we draw here extend to the vector case with no difficulty. Now, for this far apart interaction we will assume that $g(\mathbf{r},\mathbf{r}')$ can be approximated by the product of two functions that depend independently on $\mathbf{r}$ and $\mathbf{r}'$ as

$$g(\mathbf{r},\mathbf{r}') = f(\mathbf{r})h(\mathbf{r}') + e(\mathbf{r},\mathbf{r}'), \tag{4.152}$$

where $e(\mathbf{r},\mathbf{r}')$ denotes the error of the approximation. For this case, we can write the subblock as

$$[z]_{M\times M} = \{u\}_{M\times 1}\{v\}_{1\times M} + [e]_{M\times M}, \tag{4.153}$$

where

$$\{u\}_m = \int_S \psi_m(\mathbf{r})f(\mathbf{r})dS, \tag{4.154}$$

$$\{v\}_n = \int_S h(\mathbf{r}')\psi_n(\mathbf{r}')dS', \tag{4.155}$$

$$[e]_{mn} = \int_S\int_S \psi_m(\mathbf{r})e(\mathbf{r},\mathbf{r}')\psi_n(\mathbf{r}')dS'dS. \tag{4.156}$$

The matrix that is formed by the outer product $\{u\}_{M\times 1}\{v\}_{1\times M}$ is an example of a *rank-1 matrix*. This matrix can be represented efficiently with only $2M$ numbers as opposed to the full M^2 elements of the explicit matrix.

Generally, using only a single rank-1 matrix will not provide a sufficiently low error. Instead, we can seek to express the matrix using multiple rank-1 matrices as

$$[z]_{M\times M} = \sum_{r=1}^{R}\{u_r\}_{M\times 1}\{v_r\}_{1\times M} + [e]_{M\times M}, \tag{4.157}$$

which can be written more compactly as

$$[z]_{M\times M} = [u]_{M\times R}\,[v]_{R\times M} + [e]_{M\times M}. \tag{4.158}$$

This can be viewed as using more products of functions to expand $g(\mathbf{r},\mathbf{r}')$, and will naturally lower the error captured in $[e]_{M\times M}$. For this representation, we can use $2RM$ numbers to represent the full subblock of the matrix as opposed to the M^2 numbers typically needed. So long as $R \ll M$, this representation can be very efficient for both storage and the evaluation of matrix-vector products.

The standard way to compute this factorization completely is to use the *singular value decomposition (SVD)*. This is a generalization of an eigenvalue decomposition that can be applied to *any matrix*. It decomposes the matrix into a product of three matrices as

$$[z]_{M\times M} = [U]_{M\times M}[\Sigma]_{M\times M}[V]^{\dagger}_{M\times M}, \tag{4.159}$$

where $[U]$ and $[V]$ are unitary matrices that store the *singular vectors* and $[\Sigma]$ is a diagonal matrix whose entries are known as the *singular values* of $[z]$ (note, for an arbitrary matrix the matrix dimensions above can be written in a more general way). These singular values are always real, positive numbers and are stored in $[\Sigma]$ in descending order. If a matrix is rank-deficient, the values of the singular values often eventually start decreasing exponentially so that generally only a few of the singular vectors associated with the largest singular values are needed to represent $[z]_{M\times M}$ with sufficiently low error.

Computing the full SVD of a matrix is a computationally intensive task and also requires full knowledge of the matrix to be operated on. Obviously, this will not be suitable for use in a fast algorithm. The ACA provides an algorithmic way to efficiently find the compressed representation of the matrix without requiring the full computation of the matrix to be approximated.

4.13.2 Cross Approximation and Adaptive Cross Approximation

Before considering the ACA, we must first discuss the *cross approximation*, sometimes also referred to as the *skeleton approximation* of a rank-deficient matrix. The cross approximation follows an iterative procedure to gradually compute the rank-R representation of a matrix by minimizing the error matrix from step to step. In particular, we have that

$$[e]^{(k)}_{M\times M} = [z]_{M\times M} - [u]_{M\times k}[v]_{k\times M} \quad k = 0, 1, 2, \ldots, R, \tag{4.160}$$

where $[e]^{(k)}_{M\times M}$ is the error matrix at the kth iteration. The procedure begins with $k = 0$, so that the "error matrix" is just the full matrix, i.e., $[e]^{(0)}_{M\times M} = [z]_{M\times M}$. We then find the entry in $[e]^{(0)}_{M\times M}$ with the largest absolute value, which we denote as $e^{(0)}(I_1, J_1)$. We then choose the first vectors to form a rank-1 representation to be

$$u(:,1) = \frac{e^{(0)}(:,J_1)}{e^{(0)}(I_1,J_1)} \tag{4.161}$$

$$v(1,:) = e^{(0)}(I_1,:). \tag{4.162}$$

Clearly, $u(:,1)v(1,:)$ perfectly reproduces the I_1th row and J_1th column of $[e]^{(0)}_{M\times M} = [z]_{M\times M}$ so that the new error matrix

$$[e]^{(1)}_{M\times M} = [z]_{M\times M} - [u]_{M\times 1}\,[v]_{1\times M} \tag{4.163}$$

will be filled completely with zeros in the I_1th row and J_1th column. Correspondingly, the norm of the error matrix, computed as

$$||e|| = \sqrt{\sum_{m=1}^{M}\sum_{n=1}^{M}|[e]_{mn}|^2}, \tag{4.164}$$

will have been reduced (as a side note, this norm is usually known as the *Frobenius norm*). We can continue this iterative process until $||e|| < \epsilon||z||$, where ϵ is some desired tolerance level.

For the next update, we now search for the entry in $[e]^{(1)}_{M\times M}$ with the largest absolute value, which we denote as $e^{(1)}(I_2, J_2)$. We then choose as our next vectors to form a rank-1 matrix as

$$u(:, 2) = \frac{e^{(1)}(:, J_2)}{e^{(1)}(I_2, J_2)} \tag{4.165}$$

$$v(2, :) = e^{(1)}(I_2, :). \tag{4.166}$$

These will perfectly reproduce the I_2th row and J_2th column of $[e]^{(1)}_{M\times M}$, and will importantly still leave the zero entries in the I_1th row and J_1th column intact. Hence, this update procedure is guaranteed to continue to lower the error norm. We can continue to follow this procedure until we reach a desired convergence level for the representation of $[z]_{M\times M}$.

Obviously, this procedure is still not what we want for a fast algorithm as it requires us to have full access to $[z]_{M\times M}$. The ACA attempts to replicate this procedure without requiring the full computation of the matrix $[z]_{M\times M}$. It begins by selecting an arbitrary row for I_1 and then calculates $z(I_1, :)$ and also sets $v(1, :) = z(I_1, :)$. We then choose J_1 to be the column number of the largest entry in $v(1, :)$. We then calculate $z(:, J_1)$ and set $u(:, 1) = z(:, J_1)/v(1, J_1)$, which completes the first iteration.

For the next iteration, we set I_2 to be the row number of the largest entry in $u(:, 1)$ and then calculate $z(I_2, :)$. We then set $v(2, :) = z(I_2, :) - u(I_2, 1)v(1, :)$. We then find the column number J_2 as the largest entry in $v(2, :)$ and calculate $z(:, J_2)$. We conclude the second iteration by setting $u(:, 2) = [z(:, J_2) - u(:, 1)v(1, J_1)]/v(2, J_2)$. This process is repeated until we reach a desired termination condition.

To determine this termination, we ideally would need to calculate $||e^{(k)}||$ and compare it with $||z||$. This is obviously impossible for the ACA since neither of these matrices are fully computed. Instead, $||e^{(k)}||$ is approximated by the largest error contribution that has just been eliminated, i.e., $||e^{(k)}|| \approx ||u(:, k)|| \cdot ||v(k, :)||$. To estimate $||z||$, we use the norm of the approximated matrix formed by $[z]^{(k)}_{M\times M} = [u]_{M\times k}[v]_{k\times M}$.

Overall, this procedure requires $O(R^2M)$ operations to generate a rank-R matrix approximation. Storing this rank-R representation then requires memory usage of $O(RM)$. Once generated, this low-rank representation can be used in matrix-vector products to reduce the number of operations to $O(RM)$.

To actually use this approach in the MoM solution, a multilevel partitioning of the matrix is performed using the same octree-based subdivision of the geometry of interest used in the MLFMA. For distant interactions, the ACA is used to compress the matrix representation. For near interactions, the full matrix is computed and stored in a manner similar to the MLFMA. These can then be used to speed up the evaluation of matrix-vector products needed in iterative solvers.

4.13.3 Results

To see some important details about the ACA, we will consider a few important numerical results. The first set of results look at the maximum rank of subblocks of the MoM matrix for varying sizes of total problem unknowns (N) for two conditions. To keep the analysis of

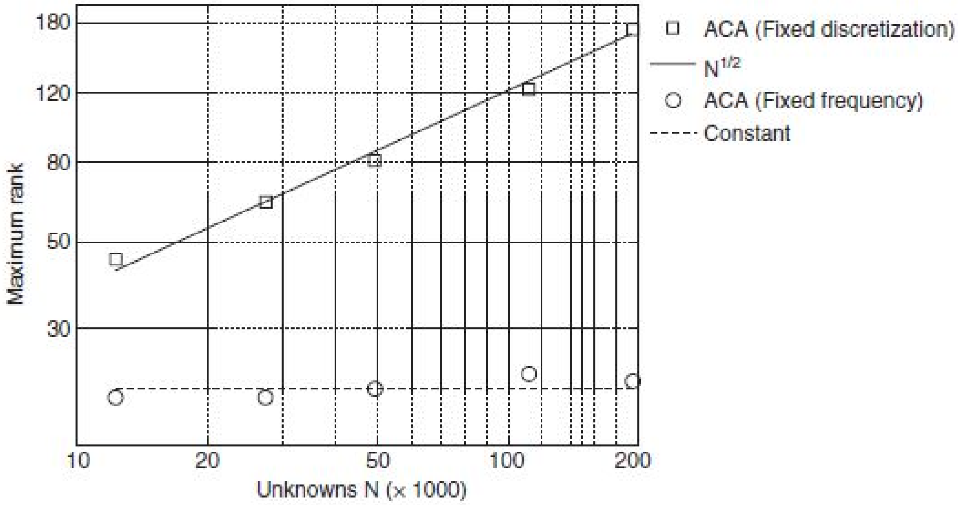


Figure 4.20: Maximum rank of the subblocks of the MoM matrix for a PEC sphere (image from [35]).

the results simple, the object studied is a PEC sphere with a radius of 1 meter. The first analysis condition fixes the analysis frequency to be 30 MHz and changes the discretization size to go from $\lambda/130$ to $\lambda/520$. The second condition keeps the discretization density fixed, but varies the analysis frequency from 600 MHz to 2.4 GHz.

The results of the maximum rank for these two cases is shown in Fig. 4.20. It is found that when the frequency is kept constant and the discretization density changes the maximum rank of the subblocks is relatively unaffected. This can be understood by the fact that the finer discretization does not really provide any "new" information to this problem since the "collective effects" between distant groups is still describable with the same number of reduced parameters compared to the underlying small current elements. In contrast to this, when the frequency is changed the electrical size of the sphere changes and so the physics between current elements at various locations along the surface also changes. In this case, it is found that the maximum rank of the subblocks is proportional to $\sqrt{N}$.

This difference in scaling of rank affects the overall efficiency of the ACA for the two cases. For the fixed frequency case, the memory and computation time scale as $O(N \log N)$. Meanwhile, the memory and computation time scale as $O(N^{4/3} \log N)$ for the case with fixed discretization density. As a result, we see that the ACA is not able to match the same performance of the MLFMA for fixed discretization density problem. The impact of these different scaling rates on the solution time and memory usage are illustrated in Fig. 4.21.

4.14 Method of Moments Project

This project covers the implementation of a computer code using the method of moments to solve problems in electromagnetics. A list of suggested project topics are included later in this document. The main deliverable for this project will be a written formal report that details the work that was completed. At a high-level, this report will cover the formulation

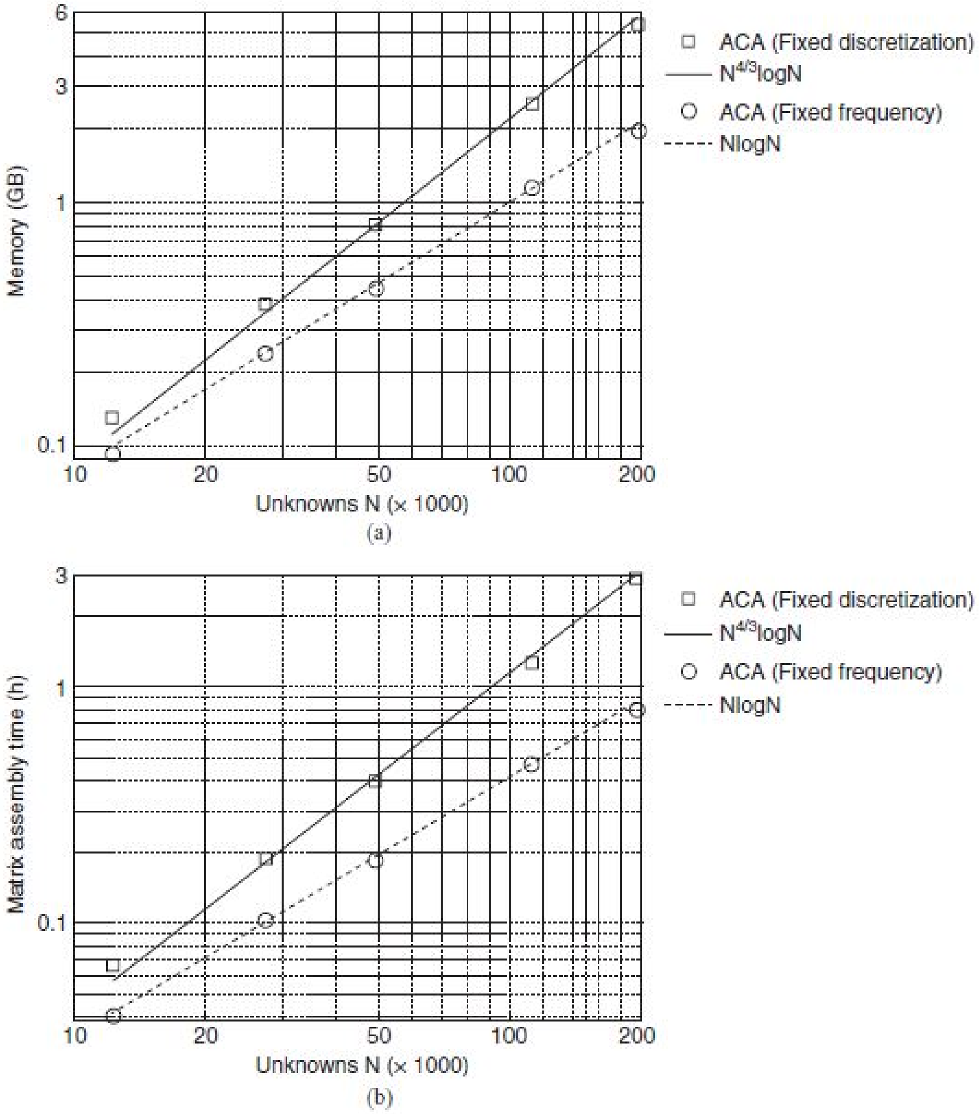


Figure 4.21: Computational complexity for the ACA algorithm applied to a PEC sphere. (a) Memory requirement and (b) computation time (images from [35]).

of the mathematical problem solved, the discretization approach used, and a discussion of the validation of the computer code via numerical results generated. A detailed grading rubric for this report is included later in this document.

4.14.1 Suggested Project Topics

1. Develop a 2D MoM program to calculate the TM_z and TE_z scattering from an infinitely-long PEC cylinder of different cross sections. Validate your results by comparing to appropriate analytical solutions for a circular cylinder with different radii (e.g., $R = \lambda$ and $R = 2\lambda$). Simulate at least one other object for both polarizations for which an analytical solution does not exist (e.g., a square cylinder) and comment on the results.

   Note: If the analytical solution you compare to is the near-field total or scattered fields **you can earn up to 5 points of extra credit for your project**.

   Note: If the analytical solution you compare to is the far-field bistatic scattering width **you can earn up to 10 points of extra credit for your project**.

2. Complete Problems 10.1 to 10.3 from [5]. These problems involve computing the capacitance of a square conducting plate using different formulas with varying levels of accuracy for the evaluation of MoM matrix elements.

3. Write a MoM code to compute the input impedance and current distribution along a thin wire antenna using the delta-gap excitation method described in Section 10.3.2 of [5]. Consider a straight dipole antenna with a length of 0.5λ and a radius of 0.001λ. Examine the effects of the discretization density on the numerical solution of the current distribution and input impedance. Compare the results to the expected theoretical value (that incorporates approximations into the analysis) of $Z_{\text{in}} = 73 + j42.5$.

   Note: If you choose this project, you can earn up to **20 points of extra credit for your project**.

4. Use the method of moments to solve one problem of interest to you (clear the problem with Dr. Roth prior to starting). Make sure to plan for some way to validate your code's performance for your selected problem.

4.14.2 Rubric

1. Title & Abstract (5 points)
   - (a) Title and abstract are concise, but informative.
   - (b) Abstract should properly convey the main information contained in the work, the methods used, and the problems studied.
2. Introduction and Conclusion (10 points)
   - (a) Introduction should discuss relevant background and history of the problem to be studied and the methods used in the work, supported by relevant references from textbooks and the literature (around 4 or 5 references is likely plenty for this report). Introduction should also finish with a paragraph discussing the organization of the remainder of the paper.
   - (b) Conclusion should succinctly summarize the content of the work and mention possible directions for further study, improvements that could be made to the numerical methods, etc.
3. Formulation & Discretization (30 points)
   - (a) Equations that are to be solved numerically are appropriately derived from a well-established starting point (e.g., Maxwell's equations).
   - (b) Assumptions or approximations of the derivation are clearly communicated.

(c) Basic process of the numerical discretization is clearly communicated for all important/distinct equations.

4. Numerical Results (45 points)

   (a) Validation data is shown to demonstrate correct implementation of the numerical method. Sufficient details on the numerical results and validation data should also be included so that someone else could conceivably implement their own tool and replicate your results. Sample items to cover would be sizes of the simulation region and any objects involved, average element size, relative permittivity and permeability of materials, etc. (Note: this is not an exhaustive list of what should be covered).

   (b) Additional numerical results are presented to show utility of the numerical method. Again, sufficient detail is provided for simulation parameters that a reader can understand the content of the simulation and recreate it themselves.

   (c) Figures are legible and aesthetically-pleasing (Matlab/Python plots are fine). Figure captions are concise, but informative. Figures are referenced and discussed appropriately within the text of the report.

   (d) Note: your code must correctly implement the numerical method to approach reaching full points in this category of the rubric.

5. Writing Style (5 points)

   (a) Grammar, word use, spelling, etc. are of an overall good quality.

   (b) Best practices for writing mathematical prose are followed (equations are treated as part of the sentence, equations are numbered, "user-friendly" references to previous equations, etc.). See "What's Wrong with these Equations?" by N. David Mermin for basic guidelines to consider.

   (c) Equations are typeset in an aesthetically-pleasing manner.

   (d) Note: if the writing style is particularly poor, additional points will be subtracted from other aspects of the report (e.g., Formulation & Discretization or Numerical Results).

6. Coding Style (5 points)

   (a) Code is formatted and organized in an easily-readable manner. Descriptive variable and function names are used as appropriate.

   (b) Sufficient comments are used to make the code more easily interpreted by another person.

Chapter 5

Concluding Remarks

5.1 Conclusion

These lecture notes have reviewed many of the fundamental concepts concerning the three major classes of CEM techniques; namely, finite difference methods, finite element methods, and the method of moments. In reality, the topics covered here only scratch the surface on the basics of modern computational electromagnetics methods. A knowledgeable reader will readily note that we have omitted many important topics with respect to more advanced subjects like preconditioning, hybrid methods, additional fast algorithms, and asymptotic methods to name just a few. To partially address these omissions, when we teach this class at Purdue we leave the final two weeks of the semester to be filled in by student presentations. Each student selects a CEM topic to independently study and then gives a "conference" style presentation to the full class to teach them about the studied topic. These presentations briefly cover many of the "missing" topics from these lecture notes, and have generally been a very popular component of the course from the student's perspective.

5.2 Final Presentation Assignment

Choose a CEM topic to independently study and develop a "conference" style presentation that you will present to the full class. Each presentation should be between 13 to 15 minutes long, followed by approximately 3 minutes of Q&A. Each presentation should discuss the background/motivation of the selected CEM topic (e.g., what problem is the technique meant to address) and convey the main important points about the topic. In almost all cases, some mathematical equations or derivations will be needed in the presentation. Only the essential points or key steps should be reviewed, with the focus on conveying the intuitive process rather than the fully explicit details. A list of suggested topics is included toward the end of this document.

5.2.1 Suggested Project Topics

1. Complex frequency shifted PML for the FDTD method

2. FDTD with irregular grids and conformal FDTD methods (note: must cover both topics)
3. Alternating direction implicit (ADI) FDTD techniques (or other implicit FDTD techniques)
4. Modeling of nonlinear electromagnetics problems with FDTD or FETD
5. Introduction to the discontinuous Galerkin time domain (DGTD) method
6. Introduction to the adaptive integral method (AIM) fast algorithm
7. Introduction to hierarchical matrices for electromagnetic integral equations
8. Low frequency breakdown of electromagnetic FEM and solution via tree-cotree decomposition
9. Low frequency breakdown of the EFIE and solution via loop-tree decomposition
10. Dense mesh breakdown of the EFIE and solution via multiplicative Calderón preconditioning
11. Formulation of integral equations for efficiently analyzing coplanar waveguide structures
12. Analysis of periodic structures using the MoM
13. Formulation and solution of volume integral equations using the MoM
14. Introduction to time domain integral equations (TDIEs)
15. Introduction to the singular value decomposition and the condition number of a matrix within the context of CEM
16. Iterative solvers and Krylov subspace methods for solving matrix equations that occur in CEM
17. Multiphysics modeling (must involve electromagnetics)
    (a) Particle-in-cell (PIC) simulations for modeling electromagnetic fields interacting with plasmas
    (b) Maxwell's equations coupled with thermal solvers
    (c) Maxwell's equations coupled with circuit solvers
    (d) Maxwell-Schrödinger semiclassical model solvers
18. Introduction to the shooting-and-bouncing ray method (must cover some details on physical optics integrals involved in these methods)
19. Hybridization of FDTD and FETD

20. Hybrid finite element-boundary integral method

21. Domain decomposition methods for FDTD, FEM, or MoM

22. Inverse scattering methods using CEM, such as the Born Iterative Method (BIM) or the Distorted Born Iterative Method (DBIM)

23. Other student-suggested and instructor-approved CEM topics

5.2.2 Rubric

1. Slides (60 points total)

   (a) Content (40 points): Technical content is informative and correctly conveys the most important points about the selected topic. Strikes the correct balance between technical detail and not overwhelming the viewer with too many equations or fine details that cannot possibly be understood in the amount of time the slide will be shown for.

   (b) Organization (10 points): Information is organized in a clear, logical way on each slide. Content flows in a sensible manner from slide to slide. Key takeaways are emphasized when appropriate on a slide.

   (c) Aesthetics (5 points): Slides are visually appealing and not too cluttered, effective graphics are used when appropriate.

   (d) Spelling and Grammar (5 points): Correct spelling and consistent grammatical style used throughout the slides.

   Note: Full sentences are often not good in bullet points on a slide.

2. Presentation (40 points)

   (a) Presentation Skills (25 points): Technical content of the speech is effectively conveyed in a clear manner. Presenter speaks in a clear voice with appropriate volume so that they can be heard. Presenter makes eye contact with everyone in the room (in-person only). Speech flows nicely with no excessively long pauses or distracting verbal fillers (uhhh, ummmm...). Presenter makes appropriate use of the slides (e.g., pointing things out when helpful), but does not just read from them.

   (b) Question and Answer (10 points): As a Presenter, is able to effectively answer questions asked about their topic. If the Presenter does not know the answer, they should mention this and then provide their best guess at an answer and their rationale for this answer. As an Audience Member, is in attendance and participates actively in the Q&A by asking insightful questions as appropriate.

   (c) Time (5 points): Presentation is finished without being excessively short (e.g., quicker than 13 minutes) or significantly over time (e.g., longer than 15 minutes).

   Note: Consider point deductions for going under or over the allotted time as following an exponential trend that can begin to deduct from the “Presentation Skills” category if it becomes excessive.

Bibliography

[1] D. Styer, “Calculation of the anomalous magnetic moment of the electron,” 2012.

[2] S. D. Gedney, “Introduction to the finite-difference time-domain (FDTD) method for electromagnetics,” in *Synthesis Lectures on Computational Electromagnetics*. Morgan & Claypool Publishers, 2011, pp. 1–250.

[3] S. Benkler, N. Chavannes, and N. Kuster, “A new 3-D conformal PEC FDTD scheme with user-defined geometric precision and derived stability criterion,” *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 6, pp. 1843–1849, 2006.

[4] A. Christ, J. Frohlich, and N. Kuster, “Correction of numerical phase velocity errors in nonuniform FDTD meshes,” *IEICE Transactions on Communications*, vol. 85, no. 12, pp. 2904–2915, 2002.

[5] J.-M. Jin, *Theory and Computation of Electromagnetic Fields*. John Wiley & Sons, 2011.

[6] C. J. Ryu, A. Y. Liu, W. E. I. Sha, and W. C. Chew, “Finite-difference time-domain simulation of the Maxwell–Schrödinger system,” *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, vol. 1, pp. 40–47, 2016.

[7] Wikipedia contributors, “Finite-difference time-domain method — Wikipedia, the free encyclopedia,” 2024, [Online; accessed 6-May-2024]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Finite-difference_time-domain_method&oldid=1221966521

[8] J.-P. Berenger, “A perfectly matched layer for the absorption of electromagnetic waves,” *Journal of Computational Physics*, vol. 114, no. 2, pp. 185–200, 1994.

[9] W. C. Chew and W. H. Weedon, “A 3D perfectly matched medium from modified Maxwell’s equations with stretched coordinates,” *Microwave and Optical Technology Letters*, vol. 7, no. 13, pp. 599–604, 1994.

[10] Z. S. Sacks, D. M. Kingsland, R. Lee, and J.-F. Lee, “A perfectly matched anisotropic absorber for use as an absorbing boundary condition,” *IEEE Transactions on Antennas and Propagation*, vol. 43, no. 12, pp. 1460–1463, 1995.

[11] S. D. Gedney, "An anisotropic perfectly matched layer-absorbing medium for the truncation of FDTD lattices," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 12, pp. 1630–1639, 1996.

[12] R. M. Joseph and A. Taflove, "FDTD Maxwell's equations models for nonlinear electrodynamics and optics," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 364–374, 1997.

[13] D. F. Kelley and R. J. Luebbers, "Piecewise linear recursive convolution for dispersive media using FDTD," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 6, pp. 792–797, 1996.

[14] R. J. Luebbers and F. Hunsberger, "FDTD for Nth-order dispersive media," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 11, pp. 1297–1301, 1992.

[15] M. Okoniewski, M. Mrozowski, and M. A. Stuchly, "Simple treatment of multi-term dispersion in FDTD," *IEEE Microwave and Guided Wave Letters*, vol. 7, no. 5, pp. 121–123, 1997.

[16] K. Cools, F. P. Andriulli, D. De Zutter, and E. Michielssen, "Accurate and conforming mixed discretization of the MFIE," *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 528–531, 2011.

[17] T. E. Roth and W. C. Chew, "Stability analysis and discretization of A-Φ time domain integral equations for multiscale electromagnetics," *Journal of Computational Physics*, vol. 408, p. 109102, 2020.

[18] ——, "Potential-based time domain integral equations free from interior resonances," *IEEE Journal on Multiscale and Multiphysics Computational Techniques*, 2021.

[19] D. M. Pozar, *Microwave Engineering.* John Wiley & Sons, 2011.

[20] J.-M. Jin, *The Finite Element Method in Electromagnetics.* John Wiley & Sons, 2015.

[21] J. Liu, J.-M. Jin, E. K. Yung, and R. S. Chen, "A fast, higher order three-dimensional finite-element analysis of microwave waveguide devices," *Microwave and Optical Technology Letters*, vol. 32, no. 5, pp. 344–352, 2002.

[22] S.-H. Lee and J.-M. Jin, "Adaptive solution space projection for fast and robust wideband finite-element simulation of microwave components," *IEEE Microwave and Wireless Components Letters*, vol. 17, no. 7, pp. 474–476, 2007.

[23] D. Jiao and J.-M. Jin, "A general approach for the stability analysis of the time-domain finite-element method for electromagnetic simulations," *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 11, pp. 1624–1632, 2002.

[24] ——, "Time-domain finite-element simulation of cavity-backed microstrip patch antennas," *Microwave and Optical Technology Letters*, vol. 32, no. 4, pp. 251–254, 2002.

[25] Wikipedia contributors, “Types of mesh — Wikipedia, the free encyclopedia,” 2024, [Online; accessed 6-May-2024]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Types_of_mesh&oldid=1211619868

[26] H. Fahs, S. Lanteri, and F. Rapetti, “A hp-like discontinuous galerkin method for solving the 2D time-domain Maxwell’s equations on non-conforming locally refined triangular meshes,” *RR-6162, INRIA*, 2007.

[27] D. Vartziotis, J. Wipper, and M. Papadrakakis, “Improving mesh quality and finite element solution accuracy by GETMe smoothing in solving the Poisson equation,” *Finite Elements in Analysis and Design*, vol. 66, pp. 36–52, 2013.

[28] G. S. Warren and W. R. Scott, “Numerical dispersion of higher order nodal elements in the finite-element method,” *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 3, pp. 317–320, 1996.

[29] W. C. Chew, *Waves and Fields in Inhomogeneous Media.* IEEE Press, 1995.

[30] J.-C. Nédélec, *Acoustic and Electromagnetic Equations: Integral Representations for Harmonic Problems.* Springer Science & Business Media, 2001.

[31] S. Rao, D. Wilton, and A. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409–418, 1982.

[32] A. Buffa and S. Christiansen, “A dual finite element complex on the barycentric refinement,” *Mathematics of Computation*, vol. 76, no. 260, pp. 1743–1769, 2007.

[33] P. Yla-Oijala and M. Taskinen, “Calculation of CFIE impedance matrix elements with RWG and $n\times$RWG functions,” *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 8, pp. 1837–1846, 2003.

[34] J. M. Song, C. C. Lu, W. C. Chew, and S. W. Lee, “Fast Illinois solver code (FISC),” *IEEE Antennas and Propagation Magazine*, vol. 40, no. 3, pp. 27–34, 1998.

[35] K. Zhao, M. N. Vouvakis, and J.-F. Lee, “The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 4, pp. 763–773, 2005.