GENERALIZING THE TEMPEST FDTD ELECTROMAGNETIC SIMULATION PROGRAM

by

Tom Pistor

Memorandum No. UCB/ERL M97/52

1 June 1997

GENERALIZING THE TEMPEST FDTD ELECTROMAGNETIC SIMULATION PROGRAM

by

Tom Pistor

Memorandum No. UCB/ERL M97/52

1 June 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

Generalizing the TEMPEST FDTD Electromagnetic Simulation Program

by

Tom Pistor

University of California at Berkeley

Professor Andrew R. Neureuther, Research Advisor

The finite-difference time-domain electromagnetic simulation program TEMPEST has been generalized and made more effective for analysis of 3D topography scattering effects in optical lithography. The Perfectly Matched Layer boundary condition for the truncation of finite-difference time-domain simulation domains has been added to TEMPEST allowing the simulation of isolated topographies. This addition required the generalization and improvement of several features of the TEMPEST program. The most notable generalization is the addition of the ability to solve different updating equations for different nodes. This allows more memory and computational resources to be devoted to nodes representing complicated materials (such as PML) and not wasted on nodes where they are not required. The updating equations for various materials are compared in terms of their computational intensities (required memory/node and computations/node/update) and it is concluded that significant memory and time savings are achieved. The domain excitation procedure has been modified to allow multiple point and plane sources while the convergence checking algorithm has been improved by considering the entire domain rather than a single plane of nodes thus reducing the possibility of false convergences. Various test simulations were run to demonstrate the effectiveness of the PML boundary condition and the TEMPEST program in lithography applications.

Acknowledgments

The author would like to thank Professor Andrew R. Neureuther and Dr. Robert Socha at The University of California at Berkeley and Professor Andreas C. Cangellaris at the University of Arizona.

This research was supported in part by the California State MICRO Program (Grant #95-113), the Semiconductor Research Corporation (95-LC-712-MC500), the SRC/ DARPA Lithography Network (SRC 96-LL6-460/MDA 972-97-1-0010) and by the UC Regents Fellowship.

Table of Contents

CHAPTER 1

Introduction

1.1 TEMPEST

TEMPEST is a computer program written and developed at Berkeley to simulate electromagnetic scattering problems seen in photolithography.[1][2][3] These problems include reflective notching from wafer topography, mask simulation, optical imaging, dynamic resist bleaching and optical inspection. The Maxwell equations are solved by the finite-difference time-domain (FDTD) method similar to that proposed by Yee[4].

The FDTD method involves spacially and temporally discretizing the simulation domain and consequently requires large amounts of memory and processor time. The original versions of TEMPEST (versions 1 and 2) solved only two dimensional periodic structures and ran on the massively parallel computer architecture of the Thinking Machines Corp. Connection Machine.[1][2] Later developments enabled full three dimensional simulation (version 3) and portation of the code to the (more convenient) workstation architecture (version 4).[3] But, lacking in all these versions is the capability to simulate fully isolated structures.

Simulation of isolated topographies requires the use of absorbing boundary conditions on all six faces of the three dimensional rectangular simulation domain. A serious drawback of past versions of TEMPEST is that the user cannot place absorbing boundary conditions on all six faces of the domain (only at the top and bottom) and thus only structures periodic in the x and y directions could be simulated. A secondary drawback of the past versions of TEMPEST is that the boundary conditions available are not the most modern. Recently, a new and promising boundary condition, PML, has been invented[5] which introduces less reflections for a greater range of angles of incidence. The primary motive for the development of TEMPEST 5.0 was addressing problems in reflective notching which required significant improvements such as the implementation of this new boundary condition.

Implementation of the PML boundary condition necessitated the generalization and reformulation of many parts of TEMPEST. As a result, many "side effect" improvements have been made along the way. The evolution of version 5.0 follows a sequence of causes and effects:

- The requirement to simulate anisotropic and magnetic materials forced TEMPEST to solve a more general form of the Maxwell Equations. (Chapter 2)
- Because the more general form of the Maxwell Equations were much more computation intensive than the ones currently being used, the ability to use different (more or less general) equations at different nodes was required. This in turn led to a memory restructuring and savings making the TEMPEST code more efficient overall.
- The use of PML forced the repositioning of the excitation plane, the convergence checking plane and the previously implemented boundary conditions. This was motivation for the generalization and reformulation of the excitation, convergence checking and boundary condition procedures in TEMPEST. (Chapters 3 and 4)

Fortunately, in following full circle through the more complex PML media, localized representations and algorithims lead to a code which requires less memory, operates more efficiently and has the added benefit of localized source excitations.

This report attempts to present both the underlying theory and the implementational aspects related to the major improvements made for TEMPEST version 5.0. It is not so much a philosophical dissertation on the FDTD simulation of electromagnetism as it is a practical document designed to explain what changes were made to the program and how they were implemented. The intended reader is perhaps the next student to make modifications to TEMPEST or the user who desires a better understanding of how TEMPEST works.

Chapter 2 will develop, from the Maxwell Equations, the discretized FDTD updating equations for various types of materials. This development is the essence of the FDTD method and hopes to equip the reader with enough insight to understand the discretization of PML Maxwell equations.

Chapter 3 addresses the issues of domain excitation and convergence checking - two aspects of TEMPEST which have been generalized and improved. Domain excitation is the mechanism whereby "light" is introduced into the domain. Convergence checking is necessary to determine if the fields have reached steady state.

Chapter 4 is devoted to boundary conditions. The various boundary conditions are organized and presented in ascending order of complexity. A large portion of the chapter is devoted to the newly implemented boundary condition PML. Much of the motivation for TEMPEST 5.0 was centered around the implementation of this boundary condition because it required TEMPEST to gain the ability to simulate a more general set of materials.

Chapter 5 gives example simulation outputs demonstrating the effectiveness of the PML boundary condition and the overall character of the TEMPEST program.

Before delving into these matters, a more practical introduction to TEMPEST presented in the next section will help the reader to better understand the logistics of the TEMPEST program and put the theory into the proper perspective.

1.2 A Practical Introduction to TEMPEST

A typical simulation with TEMPEST involves the following steps:

1. *Define Topography and Boundary Conditions.* The user must specify what materials are present, how they are situated in the simulation domain and what type of boundary conditions to use at the edges of the domain. Typical materials used in photolithography simulations are silicon, resist, glass, chrome and aluminum. Boundary conditions can be periodic, symmetric, perfectly conducting or absorbing.
2. *Specify domain excitation.* Here is where the user tells TEMPEST what type of electromagnetic radiation is present in the simulation. This is typically a plane wave or an image impinging upon the topography from above.
3. *Specify Other Run-Time Parameters.* TEMPEST needs to know what accuracy it is trying to achieve, minimum simulation time, maximum simulation time, what format to output the data, etc.
4. *Define TEMPEST Output.* The user must tell TEMPEST what part of the simulation domain needs to be output to a file on disk. Example outputs might be:

   i) The steady state electric field in the xy-plane at z=1um,

   ii) The magnetic fields in the zx-plane at y = 0um after three cycles.

   iii) The time evolution of the electric field at x=3.2um, y=2um, z=1um.
5. *Run the simulation.* The time it takes TEMPEST to find the steady state fields is dependent on many things (number of nodes in the domain, time step, topography etc.). Typical simulation times range from a few minutes for simple 2D topographies to several hours for large 3D topographies with a few million nodes. (Based on a 200MHz Ultra SPARC 2 workstation.)
6. *Interpret the output data.* Postprocessing of the output data is often needed to correctly interpret it. Examples of postprocessing are normalization of the fields and propagation of the fields at the bottom of the simulation domain to an image plane.

The information specified in the first four steps is placed inside a TEMPEST input file. An example input file is given below and corresponds to the topography and simulation results depicted in Figure 1.

```
/* Example Simulation May 1/97 */

x_node 61 y_node 1 z_node 122
x_dim .84y_dim .014 z_dim 1.68

wavelength 0.365
plane_source xy node 0 60 0 0 109 0 1 0 1 0 uniform 0 0
```

```
rectangle position 0 .84 0 .014 0 1.68 index 1 0
rectangle node 0 20 0 0 40 50 dispersive 1.62 2.17
rectangle node 40 60 0 0 40 50 dispersive 1.62 2.17
rectangle node 0 60 0 0 0 0 black_matter
rectangle node 0 60 0 0 121 121 black_matter
rectangle node 0 60 0 0 1 10 pml 0 0 -1 1 1 0 0
rectangle node 0 60 0 0 111 120 pml 0 0 1 1 1 0 0

plot ey nonsteady 1.00 node 0 121 0 0 0 60 intro.1.i
plot ey nonsteady 4.00 node 0 121 0 0 0 60 intro.4.i
plot ey nonsteady 5.00 node 0 121 0 0 0 60 intro.5.i
plot ey steady 0.00 node 0 121 0 0 0 60 intro.i
plot ey steady 0.25 node 0 121 0 0 0 60 intro.q
plot block node 0 121 0 0 0 60 intro.blk
```

How does the TEMPEST program arrive at a solution to the Maxwell Equations? As previously mentioned, TEMPEST uses the Finite-Difference Time-Domain (FDTD) method. "Finite-Difference" means that the Maxwell Equations are "discretized" by replacing derivatives with finite differences and "Time-Domain" means that the time domain (as opposed to the frequency domain) representation of the equations is discretized.

Effectively, the three dimensional rectangular simulation domain is broken up into a three dimensional lattice of nodes. The nodes are distributed evenly throughout the domain with a spacing of Δx. (one node per volume of Δx^3). Each node holds values representative of the electromagnetic field components throughout the nodal volume. The time axis is broken up into small "time steps", Δt. At each time step, each node must be "updated" with a predetermined "updating equation". Typically $\Delta x = \lambda/15$ and Δt is of the order $\Delta x/c/\mathrm{sqrt}(3)$ implying approximately 30 time steps per cycle of the field.

As shown in the Figure 1., the light energy propagates away from the source as time progresses. The wavefronts impinge upon the topography and reflect, diffract and transmit. As the simulation continues, eventually the fields reach a steady state where the fields no longer change from cycle to cycle at which time the fields are said to have converged and the simulation is finished.

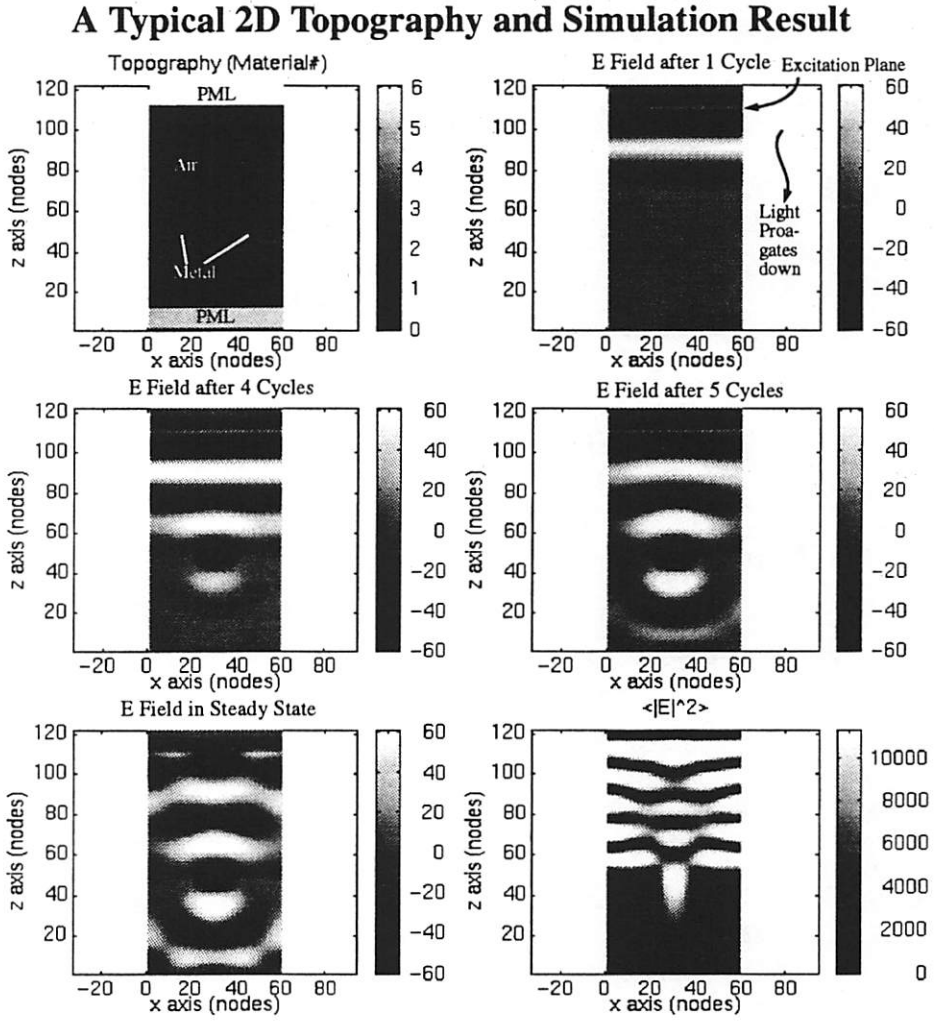

Figure 1. At the top left is the topography (or more specifically the material #). It consists of mostly air, PML at the top and bottom, and a small metal structure with a hole in the middle such that light can get through. The remaining figures are field plots. The instantaneous field plots show the light waves propagating down from the source (near the top) and impinging upon the metal structure. The bottom two plots are the steady state instantaneous electric field and the squared time averaged electric field amplitude. Note the standing waves over the metal and the diffraction of the light through the hole.

CHAPTER 2

Updating Equations

The updating equations are the equations used to recalculate, or "update", the field values at each node at each time step. They are the numerical embodiment of the Maxwell equations and describe the propagation of light from each node to its neighbors. Nodes which represent different materials require different updating equations. TEMPEST 4.0 has the capability to update nodes which correspond to linear, isotropic, non-magnetic, materials which can be dispersive or non-dispersive.

The PML material (Chapter 4), however, is an anisotropic, dispersive, magnetic material which cannot be simulated with TEMPEST 4.0. Because an ultimate goal of TEMPEST 5.0 is to incorporate the ability to simulate the PML material, the updating equations for anisotropic, magnetic, dispersive materials must be derived and programmed into the new version of TEMPEST.

This chapter discusses various materials and their associated updating equations. First, the Maxwell Equations for the simplest case, isotropic non-dispersive materials, are presented along with their standard finite difference discretized forms. Subsequently, the equations for more general materials (dispersive and anisotropic) are presented and compared in terms of computational intensity. The last section pertains to the implementation of these equations in TEMPEST 5.0 and how the various degrees of computational requirements are handled.

2.1 Basic Finite Difference Equations

Most of the concepts involved in discretizing the Maxwell Equations can be demonstrated by considering the simplest class of materials - linear, isotropic, non-magnetic, non-dispersive - and deriving the corresponding discretized equations.

2.1.1 The Maxwell Equations

The continuous form of the Maxwell Equations for linear, isotropic, non-magnetic, non-dispersive materials are written:

$$\nabla\times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\nabla\times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J}_e$$

$$\nabla\bullet \vec{D} = \rho$$

$$\nabla\bullet \vec{B} = 0 \quad , \qquad \text{(EQ 1)}$$

where $\vec{J}_e$ is the electric current density, $\vec{E}$ and $\vec{H}$ are the electric and magnetic field strengths respectively, $\vec{D}$ and $\vec{B}$ are the electric and magnetic flux densities respectively and ρ is the electric charge density.

The following two constitutive relations and current relation also apply:

$$\vec{B} = \mu(\vec{r})\vec{H}$$

$$\vec{D} = \varepsilon(\vec{r})\vec{E} \qquad \text{(EQ 2)}$$

$$\vec{J}_e = \sigma(\vec{r})\vec{E}. \qquad \text{(EQ 3)}$$

Here it is assumed that the materials involved are non-dispersive (time invariant) but the material properties may, however, be spacially varying.

It is easy to show that only the first two equations of (EQ 1) need be solved for situations where static fields are of no concern (ie. in scattering and imaging situations). Taking the divergence of the curl equations and using the fact that the divergence of the curl is zero:

$$0 = -\nabla\bullet \frac{\partial \vec{B}}{\partial t}$$

$$0 = \nabla\bullet \left(\frac{\partial \vec{D}}{\partial t} + \vec{J}_e\right) \qquad \text{(EQ 4)}$$

Employing the charge conservation property, $\nabla\bullet \vec{J}_e + \frac{\partial \rho}{\partial t} = 0$, leads to:

$$\nabla\bullet \vec{B} = const1(\vec{r})$$
$$\nabla\bullet \vec{D} = \rho + const2(\vec{r}) \qquad \text{(EQ 5)}$$

It is clear that the imposition of the divergence equations of (EQ 1) has no further consequence to the fields other than pinning down the constant (time invariant) field components of (EQ 5). It suffices to solve only the curl equations and this is what is done in TEMPEST.

2.1.2 Staggered Grid and Leap-Frog Technique

The method of finite differences can be applied to solve the Maxwell curl equations. In this method, derivatives are replaced by finite differences and the resulting equations are said to be 'discretized'. But before discretizing the equations, one must consider where to place the discrete points in space and in time.

To discretize the time axis, TEMPEST uses what's called the 'leapfrog' technique and this works as follows: The E fields are calculated at integer time steps (ie n=0,1,2,3,...) while the H fields are calculated at integer-plus-one-half time steps (ie n=0.5,1.5,2.5,...). The E field is updated with the latest H field calculated a half time step earlier and likewise for the H field. The program flow chart with the leapfrog updating scheme is shown in Figure 2.

As for the spacial discretization, TEMPEST uses a staggered grid[4] where each of the six field components E_x, E_y, E_z, H_x, H_y and H_z reside at different specially chosen positions as illustrated in Figure 3.

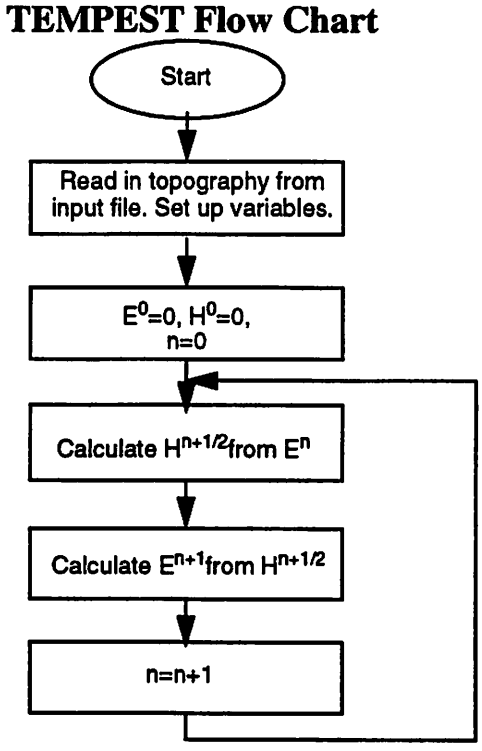

Figure 2. Flow chart for the TEMPEST program detailing the leapfrog updating scheme. Note that the electric field is evaluated at integer time steps, whereas the magnetic field is evaluated at integer-plus-one-half time steps.

This arrangement[1] may seem unorthodox at first but its convenience is seen when one examines the curl equations in component form as is done in the next section.

2.1.3 Finite Difference Discretized Equations

Now that space and time have been discretized, the curl equations in their discrete, finite difference form can be derived.

Consider a single component of the magnetic field curl equation:

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \varepsilon \frac{\partial E_z}{\partial t} + \sigma E_z \qquad \text{(EQ 6)}$$

This equation will be used to find the updating equation for the E_z field component. Note carefully the location of the E_z field component in the staggered grid (Figure 3.). The task is to use (EQ 6) to help find $E_z^{n+1}[i,j,k]$ in terms of its value at the previous time step and its nearest neighbors:

(EQ 7)

$$\{H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right], H_x^{n+\frac{1}{2}}\left[i,j-\frac{1}{2},k\right], H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right], H_y^{n+\frac{1}{2}}\left[i-\frac{1}{2},j,k\right], E_z^n[i,j,k]\}$$

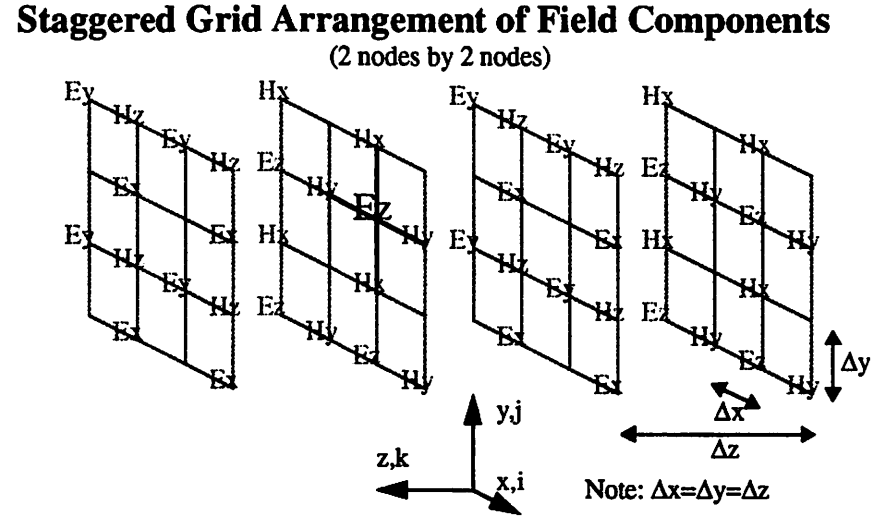

Figure 3. Note that each field component is surrounded by precisely those components needed to calculate the curl component (of its dual field) required by its updating equation.

1. Note that many other discretization schemes are possible[6]. For example, one could substitute one curl equation into the other obtaining a single (wave) equation in either E or H for which the neither the leap-frog nor the staggered grid schemes would be convenient. Although there would only be a single field to solve for, one would have to deal with second order derivatives in time and space.

As seen in the figure the nearest neighbor magnetic field components are exactly those required for a finite difference approximation to the derivatives in LHS of (EQ 6) at the (n+1/2)th time step and nodal position [i,j,k]. This is why the staggered grid is well suited to the Maxwell curl equations. Replacing spacial derivatives in (EQ 6) with their finite differences:

(EQ 8)

$$\frac{H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right]-H_x^{n+\frac{1}{2}}\left[i,j-\frac{1}{2},k\right]}{\Delta x}-\frac{H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right]-H_y^{n+\frac{1}{2}}\left[i-\frac{1}{2},j,k\right]}{\Delta x}\approx\varepsilon\frac{\partial E_z}{\partial t}+\sigma E_z\Big|_{t=(n+\frac{1}{2})\Delta t,\,\vec{r}=\Delta x[i,j,k]}$$

Now the RHS must be discretized in time. An approximation is needed for the RHS at the (n+1/2)th time step. Intuitively:

$$RHS\approx\varepsilon\frac{(E_z^{n+1}[i,j,k]-E_z^{n}[i,j,k])}{\Delta t}+\sigma\frac{(E_z^{n+1}[i,j,k]+E_z^{n}[i,j,k])}{2},\qquad\text{(EQ 9)}$$

where $E_z^{n+1}[i,j,k]$ and $E_z^{n}[i,j,k]$ have been averaged to approximate $E_z^{n+0.5}[i,j,k]$. Now together (EQ 8) and (EQ 9) can be solved for $E_z^{n+1}[i,j,k]$ in terms of known (previously calculated) quantities. Similar procedures can be followed for the remaining five field components yielding the updating equations for **isotropic, non-magnetic non-dispersive materials**:

(EQ 10)

$$E_x^{n+1}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right]=\alpha\left(E_x^{n}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right]\right)+\backslash$$
$$\beta\left(H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right]-H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k+1\right]+H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right]-H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}\right]\right)$$

$$E_y^{n+1}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right]=\alpha\left(E_y^{n}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right]\right)+\backslash\qquad\textit{(6 flops)}$$
$$\beta\left(H_z^{n+\frac{1}{2}}\left[i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right]-H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right]+H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k+1\right]-H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right]\right)$$

$$E_z^{n+1}[i,j,k]=\alpha\left(E_z^{n}[i,j,k]\right)+\backslash\qquad\textit{(6 flops)}$$
$$\beta\left(H_x^{n+\frac{1}{2}}\left[i,j-\frac{1}{2},k\right]-H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right]+H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right]-H_y^{n+\frac{1}{2}}\left[i-\frac{1}{2},j,k\right]\right)\qquad\textit{(6 flops)}$$

where, **(EQ 11)**

$$H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right] = H_x^n\left[i,j+\frac{1}{2},k\right] + \backslash$$
$$\Gamma\left(E_y^n\left[i+1,j+\frac{1}{2},k-\frac{1}{2}\right] - E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] + E_z^n[i,j+1,k] - E_z^n[i,j,k]\right) \quad \textit{(5 flops)}$$

$$H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] = H_y^n\left[i+\frac{1}{2},j,k\right] + \backslash$$
$$\Gamma\left(E_z^n[i,j,k] - E_z^n[i+1,j,k] + E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] - E_x^n\left[i+\frac{1}{2},j,k-\frac{1}{2}\right]\right) \quad \textit{(5 flops)}$$

$$H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] = H_z^{n-\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] + \backslash \quad \textit{(5 flops)}$$
$$\Gamma\left(E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] - E_x^n\left[i+\frac{1}{2},j+1,k+\frac{1}{2}\right] + E_y^n\left[i+1,j+\frac{1}{2},k+\frac{1}{2}\right] - E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right]\right)$$

$$\alpha = \frac{2\varepsilon - \sigma\Delta t}{2\varepsilon + \sigma\Delta t}$$

$$\beta = \frac{\Delta t}{\Delta x}\cdot\frac{2}{2\varepsilon + \sigma\Delta t}$$

$$\Gamma = -\frac{\Delta t}{\mu\Delta x}$$

(EQ 12)

Note that for these updating equations, TEMPEST must store 6 field components per node (3 magnetic and 3 electric) and perform 33 FLOPS per iteration per node.

The outlined scheme is shown to be stable (ie. the solution doesn't blow up) in [3] for situations where $|Re(n_{complex})| \geq |Im(n_{complex})|$, $n_{complex}$ being the complex refractive index of the material[1]. When this criterion is not met, for example in metals, a different updating scheme must be used. This is the subject of the next section.

2.2 Dispersive Materials

When the material being simulated has $|Re(n_{complex})| \leq |Im(n_{complex})|$, different updating equations must be used. Stable schemes for this situation are discussed in [3] and [13]. The updating equations used in TEMPEST for dispersive materials involve the introduction of a secondary field variable $\vec{D}$ and its associated updating equation. Here are the updating equations valid for **isotropic, non-magnetic, dispersive materials:**

1. $n_{complex} = n - jk = \sqrt{\varepsilon_r + \frac{\sigma}{j\omega\varepsilon_0}}$

$$D_x^{n+1}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] = C_1 D_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] + C_2 E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] \quad \textit{(4 flops)}$$

$$D_y^{n+1}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] = C_1 D_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] + C_2 E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] \quad \textit{(4 flops)}$$

$$D_z^{n+1}[i,j,k] = C_1 D_z^n[i,j,k] + C_2 E_z^n[i,j,k] \quad \textit{(4 flops)} \qquad \textbf{(EQ 13)}$$

(EQ 14)

$$E_x^{n+1}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] = C_3\left(E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] + D_x^{n+1}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right]\right) + \backslash$$

$$C_4\left(H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] - H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k+1\right] + H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] - H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}\right]\right)$$

(7 flops)

$$E_y^{n+1}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] = C_3\left(E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] + D_y^{n+1}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right]\right) + \backslash$$

$$C_4\left(H_z^{n+\frac{1}{2}}\left[i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] - H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] + H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k+1\right] - H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right]\right)$$

(7 flops)

$$E_z^{n+1}[i,j,k] = C_3\left(E_z^n[i,j,k] + D_z^{n+1}[i,j,k]\right) + \backslash$$

$$C_4\left(H_x^{n+\frac{1}{2}}\left[i,j-\frac{1}{2},k\right] - H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right] + H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] - H_y^{n+\frac{1}{2}}\left[i-\frac{1}{2},j,k\right]\right) \quad \textit{(7 flops)}$$

where,

(EQ 15)

If $\varepsilon_r >= 1$	If $\varepsilon_r < 1$
$\omega_0 = \dfrac{\varepsilon_0(\varepsilon_r - 1)\omega^2}{\sigma}$	$\omega_0 = \dfrac{\sigma}{\varepsilon_0(1-\varepsilon_r)}$
$C_1 = e^{-\omega_0 \Delta t}$	$C_1 = e^{-\omega_0 \Delta t}$
$C_2 = (\varepsilon_r - 1)\left(1+\dfrac{\omega^2}{\omega_0^2}\right)(1-C_1)^2$	$C_2 = (\varepsilon_r - 1)\left(1+\dfrac{\omega^2}{\omega_0^2}\right)(1-C_1)^2$
$C_3 = \dfrac{1}{1+(\varepsilon_r - 1)\left(1+\dfrac{\omega^2}{\omega_0^2}\right)(1-C_1)}$	$C_3 = \dfrac{1}{1+(1-\varepsilon_r)\left(1+\dfrac{\omega^2}{\omega_0^2}\right)(\omega_0 \Delta t - (1-C_1))}$
$C_4 = \dfrac{C_3 \Delta t}{\varepsilon_0 \Delta x}$	$C_4 = \dfrac{C_3 \Delta t}{\varepsilon_0 \Delta x}$

Note that the updating equations for the magnetic field components are the same as for the non-dispersive case, (EQ 11), since non-magnetic materials are being considered. Also note that for these updating equations, 9 field components must be stored per node and 48 flops are required per update per node (compared to 6 and 33 in the non-dispersive case).

2.3 Generalized Materials

The Maxwell Equations can be further generalized by allowing the materials involved to be anisotropic and magnetic[1]. This generalization is necessary for the simulation of the PML material which, as will be shown in Chapter 4, is an anisotropic, dispersive, magnetic, non-physical material. This section deals with the generalized Maxwell Equations and their discrete forms in an abstract sense - not specifically pertaining to PML or any other specific material.

2.3.1 Magnetic Current and Conductivity

The Maxwell Equations in a more general form are:

$$\nabla\times \vec{E} = -\frac{\partial \vec{B}}{\partial t} + \vec{J}_m$$

$$\nabla\times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J}_e$$

$$\nabla\bullet \vec{D} = \rho_e$$

$$\nabla\bullet \vec{B} = \rho_m \quad , \qquad \text{(EQ 16)}$$

where subscripts e and m are now used to differentiate between electric and magnetic current and charge. By similar arguments to those in section 2.1.1 it is only necessary to solve the curl equations.

2.3.2 Generalized Constitutive and Current Relations

Although the goal is to discretize the *time domain* equations, the constitutive and current relations (those relating $\vec{D}$ and $\vec{J}_e$ to $\vec{E}$ and $\vec{B}$, and $\vec{J}_m$ to $\vec{H}$) are most conveniently expressed and manipulated in the frequency domain. Fourier transforming the curl equations in (EQ 16) and switching to the usual complex phasor notation for the fields gives:

$$\nabla\times \tilde{E}(\omega) = -j\omega\tilde{B}(\omega) + \tilde{J}_m(\omega)$$

$$\nabla\times \tilde{H}(\omega) = j\omega\tilde{D}(\omega) + \tilde{J}_e(\omega) \quad . \qquad \text{(EQ 17)}$$

1. In this report a "magnetic" material is a material which has permeability other than that of free space and has a non-zero magnetic conductivity. This implies the existence of a magnetic current which is a non-physical phenomenon. However, just because magnetic current does not really exist does not mean that it cannot be simulated or that it is not useful.

These equations remain the same for all materials. What do change for different materials are the constitutive and current relations:

$$\begin{aligned}\tilde{D}(\omega) &= \varepsilon_0 \underline{\varepsilon}_r(\omega)\tilde{E}(\omega)\\ \tilde{B}(\omega) &= \mu_0 \underline{\mu}_r(\omega)\tilde{H}(\omega)\\ \tilde{J}_e(\omega) &= \underline{\sigma}_e(\omega)\tilde{E}(\omega)\\ \tilde{J}_m(\omega) &= \underline{\sigma}_m(\omega)\tilde{H}(\omega)\end{aligned} \quad \text{(EQ 18)}$$

Note that for the more general materials considered in this section, $\underline{\varepsilon}_r$, $\underline{\mu}_r$, $\underline{\sigma}_e$ and $\underline{\sigma}_m$ are second order frequency dependent tensors. Substitution of (EQ 18) into (EQ 17) gives:

$$\nabla\times\tilde{E}(\omega) = -j\omega\mu_0\left(\underline{\mu}_r(\omega) - \frac{\underline{\sigma}_m(\omega)}{j\omega\mu_0}\right)\tilde{H}(\omega) = -j\omega\mu_0\underline{\mu}_{complex}(\omega)\tilde{H}(\omega)$$

$$\nabla\times\tilde{H}(\omega) = j\omega\varepsilon_0\left(\underline{\varepsilon}_r(\omega) + \frac{\underline{\sigma}_e(\omega)}{j\omega\varepsilon_0}\right)\tilde{E}(\omega) = j\omega\varepsilon_0\underline{\varepsilon}_{complex}(\omega)\tilde{E}(\omega) \quad \text{, (EQ 19)}$$

where the following complex permeability and permittivity tensors have been introduced:

$$\begin{aligned}\underline{\mu}_{complex}(\omega) &= \underline{\mu}_r(\omega) - \frac{\underline{\sigma}_m(\omega)}{j\omega\mu_0}\\ \underline{\varepsilon}_{complex}(\omega) &= \underline{\varepsilon}_r(\omega) + \frac{\underline{\sigma}_e(\omega)}{j\omega\varepsilon_0}\end{aligned} \quad \text{(EQ 20)}$$

In the frequency domain, the effects of conductivity and permittivity (or permeability) are lumped together. This boils down the description of any material to two complex second order tensors $\underline{\mu}_{complex}(\omega)$ and $\underline{\varepsilon}_{complex}(\omega)$. To discretize (EQ 19), it is first necessary to transform back to the time domain. But before attempting this one must specialize to less general materials or else serious complications arise[1].

At this point the question arises: *Just how general of a material should TEMPEST be able to simulate?* The answer to this question (as far as TEMPEST 5.0 is concerned) is: *TEMPEST should be able to simulate materials at least general enough to allow the simulation of Perfectly Matched Layers (PML).*

Now the concern is: *How general of a material is PML?* It will be shown in Chapter 4 that to simulate PML it is only necessary for the tensors to be diagonal and therefore

1. Attempting to write the discrete form with the full second order tensors leads to a big mess because of the cross coupling between field components. Although it is possible to discretize these equations, it is beyond the scope of this report and not presented here.

TEMPEST need not attempt the simulation of materials with non-diagonal tensor permittivities, permeabilities and conductivities[1].

Continuing with the discretization of (EQ 19), and agreeing that the tensors are diagonal one must transform back to the time domain. How this transformation takes place will be highly dependent on the frequency dependence of $\underline{\varepsilon}_{complex}(\omega)$ and $\underline{\mu}_{complex}(\omega)$. There are four cases considered here:

Case 1 - $\underline{\varepsilon}_{complex}(\omega)$ and $\underline{\mu}_{complex}(\omega)$ are constant, diagonal and independent of ω. Transformation back to the time domain will yield equations very similar to those for the isotropic, non-dispersive case - only now, different permeabilities, permittivities and conductivities are used for each of the x,y and z field components. Here are the discrete forms:

(EQ 21)

$$E_x^{n+1}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] = \alpha_x\left(E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right]\right) + \backslash$$

$$\beta_x\left(H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] - H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k+1\right] + H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] - H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}\right]\right)$$

$$E_y^{n+1}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] = \alpha_y\left(E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right]\right) + \backslash \quad \textit{(6 flops)}$$

$$\beta_y\left(H_z^{n+\frac{1}{2}}\left[i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] - H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] + H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k+1\right] - H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right]\right)$$

$$E_z^{n+1}[i,j,k] = \alpha_z\left(E_z^n[i,j,k]\right) + \backslash \quad \textit{(6 flops)}$$

$$\beta_z\left(H_x^{n+\frac{1}{2}}\left[i,j-\frac{1}{2},k\right] - H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right] + H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] - H_y^{n+\frac{1}{2}}\left[i-\frac{1}{2},j,k\right]\right) \quad \textit{(6 flops)}$$

(EQ 22)

$$H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right] = \gamma_x H_x^n\left[i,j+\frac{1}{2},k\right] + \backslash$$

$$\Gamma_x\left(E_y^n\left[i+1,j+\frac{1}{2},k-\frac{1}{2}\right] - E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] + E_z^n[i,j+1,k] - E_z^n[i,j,k]\right) \quad \textit{(6 flops)}$$

$$H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] = \gamma_y H_y^n\left[i+\frac{1}{2},j,k\right] + \backslash$$

$$\Gamma_y\left(E_z^n[i,j,k] - E_z^n[i+1,j,k] + E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] - E_x^n\left[i+\frac{1}{2},j,k-\frac{1}{2}\right]\right) \quad \textit{(6 flops)}$$

$$H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] = \gamma_z H_z^{n-\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] + \backslash$$

$$\Gamma_z\left(E_x^n\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] - E_x^n\left[i+\frac{1}{2},j+1,k+\frac{1}{2}\right] + E_y^n\left[i+1,j+\frac{1}{2},k+\frac{1}{2}\right] - E_y^n\left[i,j+\frac{1}{2},k+\frac{1}{2}\right]\right) \quad \textit{(6 flops)}$$

1. Note that some non-diagonal materials can, through a coordinate transformation, be diagonalized but in general, with unrelated $\underline{\varepsilon}$ and $\underline{\mu}$ tensors no diagonalizing transformation exists.

where,

$$\alpha_i = \frac{2\varepsilon_0\varepsilon_{r,i} - \sigma_{e,i}\Delta t}{2\varepsilon_0\varepsilon_{r,i} + \sigma_{e,i}\Delta t}$$

$$\beta_i = \frac{\Delta t}{\Delta x} \cdot \frac{2}{2\varepsilon_0\varepsilon_{r,i} + \sigma_{e,i}\Delta t} \qquad i = \{x, y, z\}$$

$$\gamma_i = \frac{2\mu_0\mu_{r,i} - \sigma_{m,i}\Delta t}{2\mu_0\mu_{r,i} + \sigma_{m,i}\Delta t}$$

$$\Gamma_i = -\frac{\Delta t}{\Delta x} \cdot \frac{2}{2\mu_0\mu_{r,i} + \sigma_{m,i}\Delta t} \qquad \text{(EQ 23)}$$

Equations (EQ 10) to (EQ 12) are valid for **anisotropic, magnetic, non-dispersive materials.**

Case 2 - $\underline{\varepsilon}_{complex}(\omega)$ models an electrically dispersive material (such as a metal) and $\underline{\mu}_{complex}(\omega)$ is constant and independent of ω. As discussed in [3] there are different possibilities for $\underline{\varepsilon}_{complex}(\omega)$ when modelling a metal. Using the Lorentz and Debye models for permittivity (see Chapter 5 of [3]), yields the dispersive equations presented in Section 2.2.

Case 3 - $\underline{\varepsilon}_{complex}(\omega)$ and $\underline{\mu}_{complex}(\omega)$ are chosen to eliminate reflections between material interfaces. This is precisely the condition which is met by the PML material (Chapter 4) and will lead to diagonal tensors of the following form:

$$\underline{\varepsilon}_{complex}(\omega) = \varepsilon_1(\omega)\,\underline{\Delta}(\omega)$$

$$\underline{\mu}_{complex}(\omega) = \mu_1(\omega)\,\underline{\Delta}(\omega)$$

$$\underline{\Delta}(\omega) = \begin{bmatrix} a(\omega) & 0 & 0 \\ 0 & b(\omega) & 0 \\ 0 & 0 & c(\omega) \end{bmatrix}, \qquad \text{(EQ 24)}$$

where $\varepsilon_1(\omega)$ and $\mu_1(\omega)$ are frequency dependent scalars. Note the connection between $\underline{\varepsilon}_{complex}(\omega)$ and $\underline{\mu}_{complex}(\omega)$ - they differ only by a scalar. Further details regarding the exact form of the scalars and the $\underline{\Delta}(\omega)$ tensor are given in Chapter 4.

Case 4 - $\underline{\varepsilon}_{complex}(\omega)$ and $\underline{\mu}_{complex}(\omega)$ are chosen as independent dispersion equations which are the ratio of polynomial functions in $(j\omega)$ ie:

$$\underline{\varepsilon}_{complex}(\omega) = Q(j\omega) = \frac{a_0 + a_1(j\omega) + a_2(j\omega)^2 + \ldots}{b_0 + b_1(j\omega) + b_2(j\omega)^2 + \ldots}$$

$$\underline{\mu}_{complex}(\omega) = R(j\omega) = \frac{c_0 + c_1(j\omega) + c_2(j\omega)^2 + \ldots}{d_0 + d_1(j\omega) + d_2(j\omega)^2 + \ldots} \qquad \text{(EQ 25)}$$

This is the most general of the cases considered in this paper.[1] As previously mentioned, these relations must first be converted to the time domain before they can be discretized. The second equation of (EQ 19) with the first of (EQ 25) yields:

$$\begin{aligned}
&\nabla\times \tilde{H}(\omega) = j\omega\varepsilon_0 Q(j\omega)\tilde{E}(\omega)\\
&\Rightarrow (b_0 + b_1(j\omega) + b_2(j\omega)^2 + \ldots)(\nabla\times \tilde{H}(\omega))\\
&\qquad = \varepsilon_0(a_0(j\omega) + a_1(j\omega)^2 + a_2(j\omega)^3 + \ldots)\tilde{E}(\omega)\\
&\Rightarrow \left(\sum_{p=0}^{P} b_p(j\omega)^p\right)(\nabla\times \tilde{H}(\omega)) = \varepsilon_0\left(\sum_{s=0}^{S} a_s(j\omega)^{s+1}\right)\tilde{E}(\omega)\\
&\Rightarrow \left(\sum_{p=0}^{P} b_p\frac{\partial^p}{\partial t^p}\right)(\nabla\times \vec{H}(t)) = \varepsilon_0\left(\sum_{s=0}^{S} a_s\frac{\partial^{(s+1)}}{\partial t^{s+1}}\right)\vec{E}(t)
\end{aligned} \qquad \text{(EQ 26)}$$

where the last of (EQ 26) is the time domain representation. Discretization could be carried out on the last of (EQ 26) but is not because more efficient discretization schemes may exist. The scheme outlined so far is a "brute force" method and always leads to a discretized form, but, manipulations in the frequency domain can lead to more efficient discrete forms. These frequency domain manipulations depend highly on the form of $Q(j\omega)$ and must be handled on an individual basis.

2.3.3 Some Discretization Examples

Here are two examples that illustrate the discretization procedure.

Example 1: Suppose

$$\begin{aligned}
Q(j\omega) &= \varepsilon_\infty + \frac{\varepsilon_s - \varepsilon_\infty}{1 - \frac{j\omega}{\omega_0}}\\
&= \frac{\varepsilon_s\omega_0 - \varepsilon_\infty j\omega}{\omega_0 - j\omega},
\end{aligned} \qquad \text{(EQ 27)}$$

so that from (EQ 26)

$$\begin{aligned}
&\nabla\times \tilde{H}(\omega) = j\omega\varepsilon_0\frac{\varepsilon_s\omega_0 - \varepsilon_\infty j\omega}{\omega_0 - j\omega}\tilde{E}(\omega)\\
&\Rightarrow (\omega_0 - (j\omega))(\nabla\times \tilde{H}(\omega)) = (\varepsilon_0\varepsilon_s\omega_0(j\omega) - \varepsilon_0\varepsilon_\infty(j\omega)^2)\tilde{E}(\omega)\\
&\Rightarrow (\omega_0 - \frac{\partial}{\partial t})(\nabla\times \vec{H}(t)) = (\varepsilon_0\varepsilon_s\omega_0\frac{\partial}{\partial t} - \varepsilon_0\varepsilon_\infty\frac{\partial^2}{\partial t^2})\vec{E}(t)
\end{aligned}, \qquad \text{(EQ 28)}$$

1. Note that Cases 1 to 3 are just special cases of Case 4.

where the last equation in (EQ 28) is the time domain representation. To discretize, replace the time derivatives with finite differences:

(EQ 29)

$$\omega_0 (\nabla\times \vec{H})^{n+0.5} - \frac{\left(\frac{(\nabla\times \vec{H})^{n+1.5} - (\nabla\times \vec{H})^{n+0.5}}{\Delta t}\right) + \left(\frac{(\nabla\times \vec{H})^{n+0.5} - (\nabla\times \vec{H})^{n-0.5}}{\Delta t}\right)}{2}$$

$$= \varepsilon_0\varepsilon_s\omega_0\left(\frac{\vec{E}^{n+1} - \vec{E}^{n}}{\Delta t}\right) - \varepsilon_0\varepsilon_\infty \frac{\left(\frac{\vec{E}^{n+2} - 2\vec{E}^{n+1} + \vec{E}^{n}}{\Delta t^2}\right) + \left(\frac{\vec{E}^{n+1} - 2\vec{E}^{n} + \vec{E}^{n-1}}{\Delta t^2}\right)}{2}$$

which can be solved for $\vec{E}^{n+2}$. Note that this updating scheme requires the storage of three electric fields $\vec{E}^{n+1}$, $\vec{E}^{n}$, $\vec{E}^{n-1}$ and three magnetic fields $\vec{H}^{n+1.5}$, $\vec{H}^{n+0.5}$ and $\vec{H}^{n-0.5}$.

Example 2: Another approach is possible if $Q(j\omega)$ can be factored. If so, some of the factors in the numerator may cancel with some in the denominator thus simplifying the situation. Also, secondary variables may be introduced which can reduce the number of magnetic field components which must be stored (at the cost of storing extra secondary variables).

Let

$$Q(j\omega) = (1 + \frac{\sigma}{j\omega\varepsilon_0})\left(\kappa_x + \frac{\sigma_x}{j\omega\varepsilon_0}\right). \quad \text{(EQ 30)}$$

Inserting (EQ 30) into the curl equation yields:

$$\nabla\times \tilde{H}(\omega) = j\omega\varepsilon_0 (1 + \frac{\sigma}{j\omega\varepsilon_0})\left(\kappa_x + \frac{\sigma_x}{j\omega\varepsilon_0}\right)\tilde{E}(\omega)$$

$$\nabla\times \tilde{H}(\omega) = \cancel{j\omega\varepsilon_0}\left(\frac{j\omega\varepsilon_0 + \sigma}{\cancel{j\omega\varepsilon_0}}\right)\left(\kappa_x + \frac{\sigma_x}{j\omega\varepsilon_0}\right)\tilde{E}(\omega)$$

$$\nabla\times \tilde{H}(\omega) = (j\omega\varepsilon_0 + \sigma)\left(\kappa_x + \frac{\sigma_x}{j\omega\varepsilon_0}\right)\tilde{E}(\omega) \quad \text{(EQ 31)}$$

Note the cancellation. Now a secondary variable $\tilde{D}(\omega)$ is introduced and is defined as:

$$\tilde{D}(\omega) = \left(\kappa_x + \frac{\sigma_x}{j\omega\varepsilon_0}\right)\tilde{E}(\omega) \quad \text{(EQ 32)}$$

simplifying (EQ 31) to

$$\nabla\times \tilde{H}(\omega) = (j\omega\varepsilon_0 + \sigma)\tilde{D}(\omega). \quad \text{(EQ 33)}$$

Now (EQ 31) has been broken into two separate equations, (EQ 32) and (EQ 33), each of which can be discretized in time:

$$\nabla\times\vec{H} = j\omega\varepsilon_0\frac{\vec{D}^{n+1}-\vec{D}^{n}}{\Delta t}+\sigma\frac{\vec{D}^{n+1}+\vec{D}^{n}}{2} \qquad \text{(EQ 34)}$$

$$j\omega\varepsilon_0\frac{\vec{D}^{n+1}-\vec{D}^{n}}{\Delta t} = j\omega\varepsilon_0\kappa_x\frac{\vec{E}^{n+1}-\vec{E}^{n}}{\Delta t}+\sigma_x\frac{\vec{E}^{n+1}+\vec{E}^{n}}{2} \qquad \text{(EQ 35)}$$

As can be seen, (EQ 34) can be used first to update $\vec{D}$ from $\vec{H}$ and then (EQ 35) can be used to update $\vec{E}$ with $\vec{D}$. This method, sometimes called the two-step method, is similar to that used in discretizing the equations for dispersive materials and the PML material (Chapter 4).

2.4 Implementation of the Bulk Equations in TEMPEST 5.0

As noted in the previous sections, different materials require different updating equations, some of which are more computationally intensive than others. This fact is summarized in Table 1. Included in the list of materials are three types of PML materials (1D, 2D and 3D PML) which are described in more detail in Chapter 4.

Table 1. Computational Intensities for Various Materials

Type of Material	# of field components to store per node	# of FLOPS[a] per update per node
(An) Isotropic, non-dispersive, non-magnetic (case 1)	6	33
(An) Isotropic, non-dispersive, magnetic (case 1)	6	36
(An) Isotropic, electrically dispersive (Lorentz model),non-magnetic (case 2)	9	48
(An)Isotropic, electrically and magnetically dispersive (Lorentz model), magnetic (case 3)	12	66
1D PML (case 3)	12	66[b]
2D PML (case 3)	14	96[b]
3D PML (case 3)	18	96[b]
Generalized Dispersion Equations (case 4)	unlimited, depends on degree of Q(jω) and R(jω)	unlimited, depends on degree of Q(jω) and R(jω)

a. Here a FLOP is +,-,* or / involving two floating point (32 bit) numbers.

b. Does not include calculation of material parameters.

A major drawback of TEMPEST 4.0 is that all nodes are required to be updated with the same updating equations - either the Yee equations(EQ 10)-(EQ 12) or the dispersive equations (EQ 11), (EQ 13) to (EQ 15). This often leads to inefficient simulation. Take for example the case where the simulation domain contains a relatively small percentage of isotropic, electrically dispersive, non-magnetic material say 10% and the remainder of the domain is isotropic, non-dispersive, non-magnetic material (air for example). TEMPEST 4.0 will require all nodes to use the isotropic electrically dispersive, non-magnetic equations implying 9 field components/node and 48 FLOPS/update/node.

A goal of version 5.0 is to overcome this inefficiency by allowing nodes to be updated with differing updating equations. In the above example the 10% dispersive nodes will be updated with the dispersive equations while the remaining 90% non-dispersive nodes will be updated with the non-dispersive equations implying 6.3(=0.1*9+0.9*6) field components/node and 34.5(=0.1*48+0.9*33) FLOPS/update/node - a considerable savings over TEMPEST 4.0.

This ability to use different updating equations at different nodes will be further exploited when boundary conditions are introduced in Chapter 4. Some nodes, rather than being considered as materials (such as air or metal) will be considered as boundary conditions and they will have updating equations that do not correspond to any physical materials. Some of the updating equations will require several secondary variables and many FLOPS/update/node. Luckily they occupy only a small fraction of the domain, and with TEMPEST 5.0, should therefore not waste too much time or memory.

CHAPTER 3

Generalizing Domain Excitation and Convergence Checking

The bulk updating equations presented in the previous chapter have no consideration for the introduction of electromagnetic energy into the simulation domain. If one were to set all fields initially to zero, then begin updating with the bulk equations, the fields would remain at zero forever. There is nothing in those equations which say "put a light source here". This is where domain excitation comes into play.

Another, issue not considered by the updating equations is that of checking for convergence. How does TEMPEST know when enough iterations of the updating equations have been performed and when can TEMPEST say "OK, the fields have reached steady state."? This is the task of convergence checking.

3.1 Domain Excitation

The program flow chart with domain excitation is shown in Figure 4. The basic idea is that nodes which are excited get an excitation component added to them between updates. For unexcited nodes the excitation component is zero and therefore the addition is not necessary. Typically only a very small fraction of the nodes in the simulation domain (such as a single plane of nodes near the top) are excited and therefore only a small fraction of simulation time is devoted to domain excitation.

3.1.1 The Equations

The domain is excited by adding to the electric field between updates with the bulk equations:

$$\vec{E}^n \leftarrow \vec{E}^n + \vec{E}^n_{src}, \qquad \text{(EQ 36)}$$

where

(EQ 37)

$$\vec{\hat{E}}^{n}_{src} = \vec{\hat{E}}_{src}(\vec{\hat{r}}, n\Delta t) = \sum_i \vec{\hat{E}}_{ptsrc,i}(\vec{\hat{r}}, n\Delta t) + \sum_i \vec{\hat{E}}_{Zplnsrc,i}(\vec{\hat{r}}, n\Delta t) + \sum_i \vec{\hat{E}}_{Xplnsrc,i}(\vec{\hat{r}}, n\Delta t) + \sum_i \vec{\hat{E}}_{Yplnsrc,i}(\vec{\hat{r}}, n\Delta t)$$

is the source field consisting of the sum of the fields from all the sources. Note that multiple point and plane sources are allowed. The fields for each of the individual sources are written:

$$\begin{aligned}
\vec{\hat{E}}_{ptsrc,i}(\vec{\hat{r}}, t) &= Re\left(\vec{A}_{ptsrc,i}\,\delta(\vec{\hat{r}} - \vec{\hat{r}}_{ptsrc,i})\,exp(j\omega t)\right) \\
\vec{\hat{E}}_{Zplnsrc,i}(\vec{\hat{r}}, t) &= Re\left(\vec{A}_{Zplnsrc,i}(x, y)\,\delta(z - z_{Zplnsrc,i})\,exp(j\omega t)\right) \\
\vec{\hat{E}}_{Xplnsrc,i}(\vec{\hat{r}}, t) &= Re\left(\vec{A}_{Xplnsrc,i}(y, z)\,\delta(x - x_{Xplnsrc,i})\,exp(j\omega t)\right) \\
\vec{\hat{E}}_{Yplnsrc,i}(\vec{\hat{r}}, t) &= Re\left(\vec{A}_{Yplnsrc,i}(z, x)\,\delta(y - y_{Yplnsrc,i})\,exp(j\omega t)\right)
\end{aligned} \qquad \text{(EQ 38)}$$

TEMPEST Flow Chart - With Domain Excitation

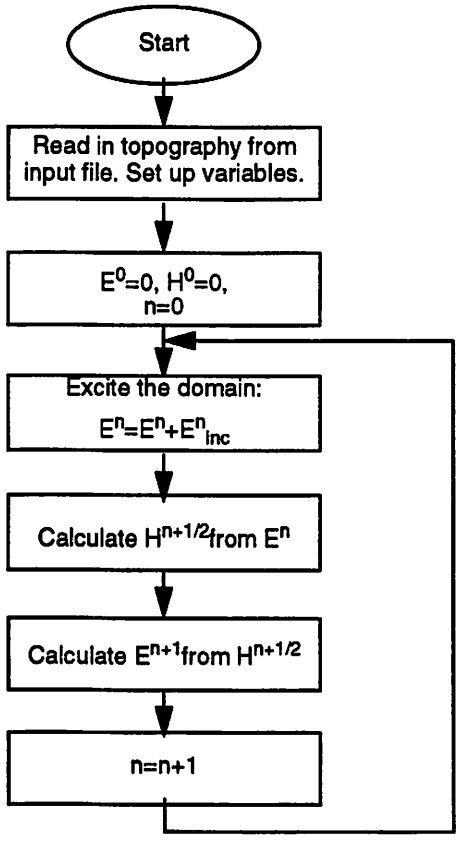

Figure 4. Program flow diagram with domain excitation included. Note that domain excitation must happen before each update.

The field information is contained in the $\vec{A}_{ptsrc,i}$, $\vec{A}_{Zplnsrc,i}$, $\vec{A}_{Xplnsrc,i}$ and $\vec{A}_{Yplnsrc,i}$ complex variables. $\vec{A}_{Zplnsrc,i}$, $\vec{A}_{Yplnsrc,i}$ and $\vec{A}_{Yplnsrc,i}$ denote planar segment sources with normals directed in the z,x and y directions respectively while $\vec{A}_{ptsrc,i}$ denotes a point source. Typically when only one planar source is used, the planar source with the z-directed normal is chosen corresponding to a source field specified in the xy-plane.

3.1.2 Some Examples

Example 1. For a point source, the magnitude, phase and polarization are contained in $\vec{A}_{ptsrc}$. A z-polarized point source with electric field $3\cos(\omega t)$ V/m corresponds to $\vec{A}_{ptsrc,i} = 3\hat{z}$.

Example 2. For a uniform plane wave of unit amplitude travelling in the z direction set $\vec{A}_{Zplnsrc,i} = 1$.

Example 3. For a uniform plane wave of amplitude 4, y-polarized electric field, travelling in the $\theta = 15°$, $\phi = 45°$ direction (see Figure 5.) with phase at (y,z)=(2,3) of $\pi/2$ radians:

$$\vec{A}_{Xplnsrc,i} = 4\hat{y}e^{j\frac{\pi}{2}(k_y(y-2)+k_z(z-3))}$$

$$k_y = k_0 \sin(\theta)\cos(\phi)$$

$$k_z = k_0 \sin(\theta)\sin(\phi) \qquad \text{(EQ 39)}$$

Example 4. For a mask or an image, TEMPEST 5.0 allows the user to load in $\vec{A}$ from a file on disk. This file would contain the magnitude and phase for each node in the excited plane.

3.2 Convergence Checking

The goal in convergence checking is to determine whether or not the fields have reached steady state. This is easily done by comparing the fields at two times one cycle (1/f seconds) apart. If they are equal (by some specified criterion) then the fields are assumed to have reached steady state. Once in steady state, no further changes to the fields are expected. The program flow chart now includes a convergence checking step and is shown in Figure 6.

3.2.1 Implementation

TEMPEST 4.0 checks for convergence by sampling an xy-plane of nodes near the top of the simulation domain[3]. The problem with this method is that in some cases the fields remain the same in the plane where convergence is being checked, but are not in steady state elsewhere in the domain. This will cause TEMPEST 4.0 to think the fields are in steady state and prematurely end the simulation.

Diagram for Example 3 Defining Angles of Propagation

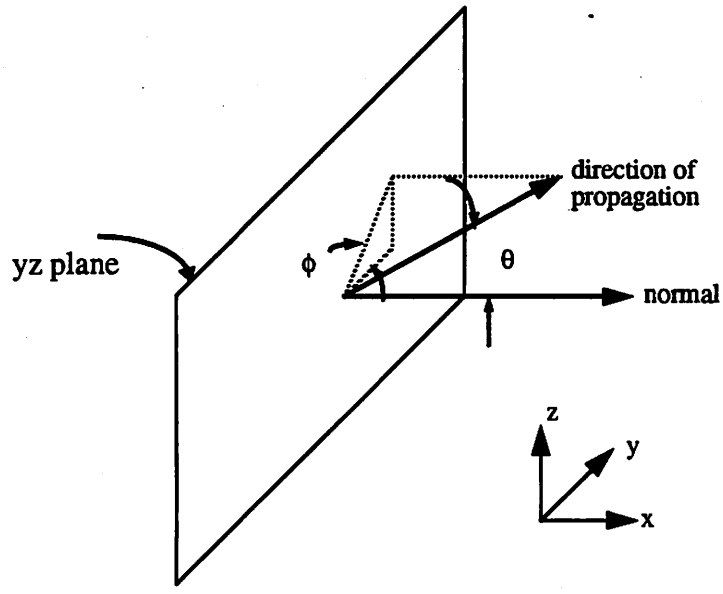

Figure 5. Note that θ is always measured from the normal (in this case the x-axis) to the plane and ϕ is always measured from the next-cyclic-permutation-of-{x,y,z}-axis (in this case the y-axis).

TEMPEST 5.0 uses a cubic grid of test points (spaced five nodes apart in x,y and z) for the convergence checking. This ensures a more uniform and complete coverage of the domain. It is much less likely that a false convergence may occur and stop the simulation before steady state has been reached.

Because the amplitude - not the instantaneous value of the field - at two different times must be compared for each converge check point, it is necessary to calculate the amplitude from the instantaneous fields at each cycle from the following formula:

$$\hat{\vec{E}}_{amp}(cT) = \sqrt{\hat{\vec{E}}\left(cT-\frac{T}{4}\right)^2 + \hat{\vec{E}}(cT)^2} \qquad c = 1, 2, 3, \ldots \qquad \text{(EQ 40)}$$

where c is the cycle number and T is the cycle period.

The formula used for comparison is the simple relative difference:

$$pterr = \frac{\hat{\vec{E}}_{amp}(cT) - \hat{\vec{E}}_{amp}((c-1)T)}{\hat{\vec{E}}_{amp}(cT) + \hat{\vec{E}}_{amp}((c-1)T)}, \qquad \text{(EQ 41)}$$

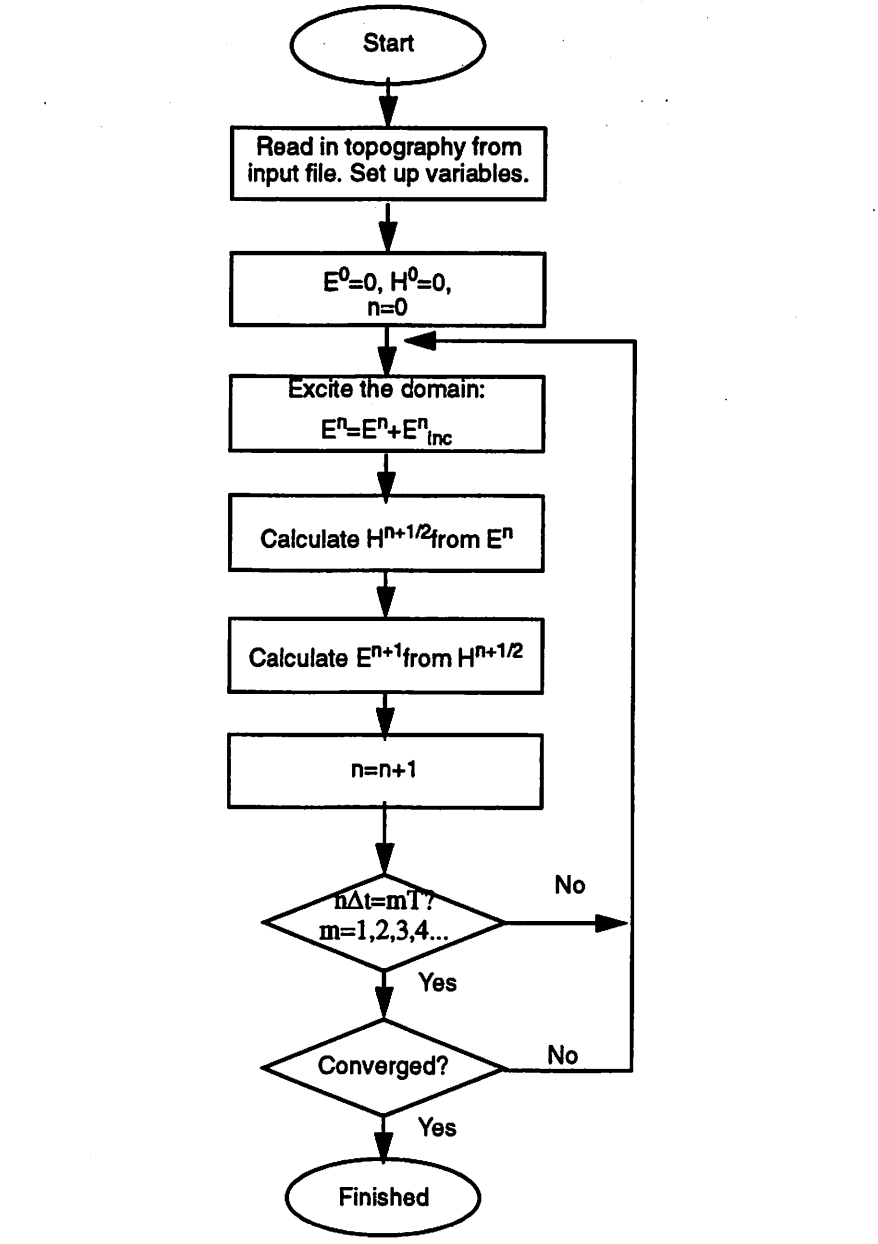

Figure 6. Program flow with convergence checking. Note that convergence is checked only at time steps which correspond to the beginning of a wave cycle.

unless the denominator is deemed extremely small in which case the ratio is not calculated. Then the error at each point is compared to mre, the maximum relative error (usually set to 0.2) and quantized to a zero or one which is summed over all test points to calculate the total error:

$$PTERR = \begin{cases} 0, pterr \leq mre \\ 1, pterr > mre \end{cases}$$

$$TOTERR = \sum_{testpoints} PTERR \qquad \text{(EQ 42)}$$

In TEMPEST 5.0, the fields are said to have converged if TOTERR=0 for three consecutive cycles.

One important implication of the above scheme is that the field components at the required times during the cycle are available (t=0.75T,1T,1.75T,2T etc.). This puts a restriction on the allowable time steps Δt specifically:

$$\Delta t = \frac{T}{4n}, n \in I. \qquad \text{(EQ 43)}$$

CHAPTER 4

Boundary Conditions

The nodes at the outermost edges of the simulation domain are missing some of their nearest neighbors and therefore must be treated as special cases when updated with the bulk equations. How are these nodes updated? What is used in place of their nearest neighbors? These questions are answered in this chapter.

4.1 Perfect Conductor Boundary Condition

The most straightforward and simple boundary condition is to simply set the fields to zero at the boundaries. This means that the outermost nodes' updating equations become:

$$E_x^{n+1}\left[i+\frac{1}{2},j,k+\frac{1}{2}\right] = 0$$

$$E_y^{n+1}\left[i,j+\frac{1}{2},k+\frac{1}{2}\right] = 0$$

$$E_z^{n+1}[i,j,k] = 0$$

$$H_x^{n+\frac{1}{2}}\left[i,j+\frac{1}{2},k\right] = 0$$

$$H_y^{n+\frac{1}{2}}\left[i+\frac{1}{2},j,k\right] = 0$$

$$H_z^{n+\frac{1}{2}}\left[i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}\right] = 0 \qquad \text{(EQ 44)}$$

which is equivalent (if the fields were initially set to zero) to not updating the nodes at all. This turns out to be equivalent to the updating equations for a perfect conductor which is understood when one considers the fact that the fields must be zero inside a perfect conductor.

The perfect conductor boundary condition is, in general, not convenient since one doesn't want to always simulate inside perfectly conducting metal boxes.

4.2 Symmetric and Periodic Boundary Conditions

Another option is to not change the updating equations but rather use some other nodes of the domain as replacements for the missing nearest neighbors. For each of the x,y and z faces of the rectangular domain there are two choices which render sensible results:

Periodic Boundary Condition: For an edge node, choose as its nearest (missing) neighbor replacement the corresponding node on the opposite face of the domain. (Example: For a domain of 10x10x10 nodes[1], edge nodes [x,y,9] are missing their nearest neighbors at [x,y,10] so they use the nodes at [x,y,0] as their replacements.) This effectively makes the simulation domain infinite and periodic in the direction of the normal to the domain faces to which it is applied.

Symmetric Boundary Condition: For an edge node, choose as its nearest (missing) neighbor replacement the corresponding node on the nearest neighboring parallel plane in the domain. (Example: For a domain of 10x10x10, edge nodes [x,y,9] are missing their nearest neighbors at [x,y,10] so they use the nodes at [x,y,8] as their replacements. Note that now the [x,y,8] nodes are used as neighbors twice.) This makes the simulation domain infinite but now with a period of twice that of the first choice and with a symmetrical structure within each period.

The two choices are illustrated in Figure 7.

These boundary conditions are easy to implement and are useful. There are many situations where it is desirable to simulate infinite periodic domains (a grating or a planar interface for example). There are a few important things to keep in mind when using periodic or symmetric boundary conditions:

- They can be applied independently to faces with x,y and z directed normals but must be applied the same for both faces with parallel normals.
- The excitation takes on the same symmetry and periodicity as the topography which places some restrictions on the incident fields near the edges - namely that they be continuous across the boundary. Example: for plane wave incidence, a domain with dimensions {X x Y x Z} periodic in the x and y dimensions, the angles of incidence θ and ϕ (see Figure 8.) of the plane wave must be such that:

1. Nodes are numbered starting from zero.

Periodic and Symmetric Boundary Condition Effects

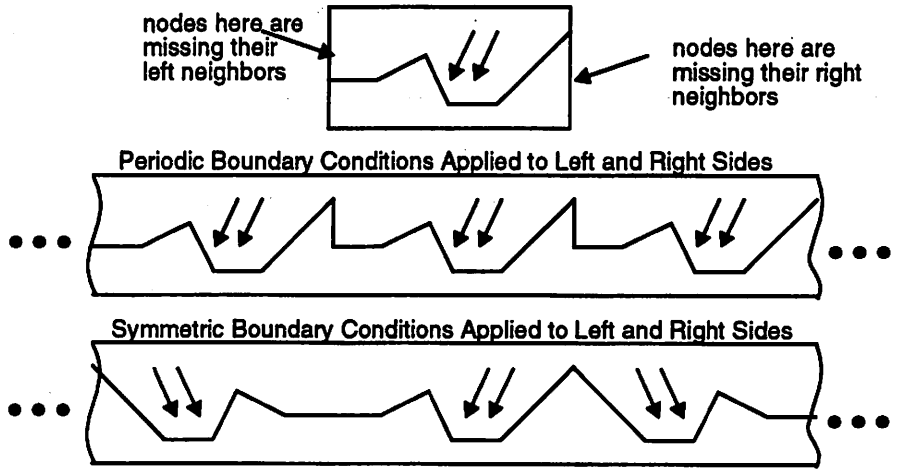

Figure 7. In the first diagram the actual (what's stored in the computer) simulation domain is depicted. The second diagram shows what is really being simulated when periodic boundary conditions are used and the third when symmetric boundary conditions are used.

Defining the Angles of Incidence for a Plane Wave

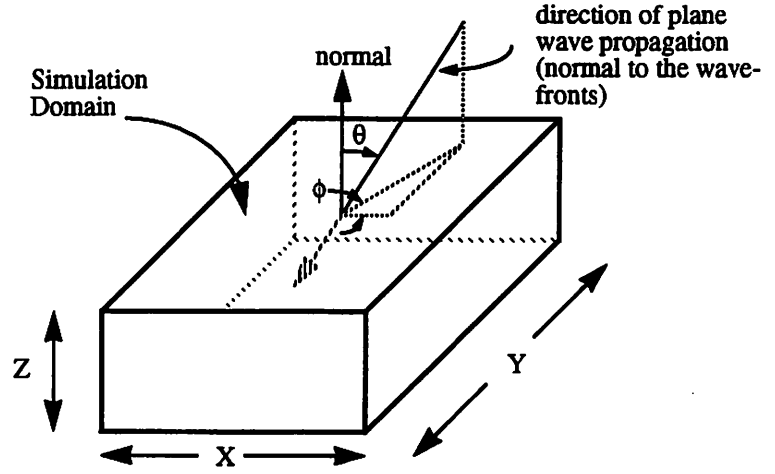

Figure 8. The angles of incidence for a plane wave, θ and ϕ, are always measured from the normal (in this case the z-axis) and the next-cyclic-permutation-of-{x,y,z} axis (in this case the x-axis).

$$k_x X = 2\pi m \qquad \{m, n\} \in I$$
$$k_y Y = 2\pi n$$

$$\Rightarrow \frac{2\pi}{\lambda} X \sin\theta\cos\phi = 2\pi m \Rightarrow \sin\theta\cos\phi = \frac{m\lambda}{X}$$

$$\Rightarrow \frac{2\pi}{\lambda} Y \sin\theta\sin\phi = 2\pi n \Rightarrow \sin\theta\sin\phi = \frac{n\lambda}{Y} \qquad \text{(EQ 45)}$$

The periodic and symmetric boundary conditions are not useful when the intention is to simulate isolated topographies. In these cases a boundary condition which absorbs the outgoing radiation is needed. The remaining boundary conditions presented in this chapter do this.

4.3 First Order Absorbing Boundary Condition

This boundary condition absorbs outgoing radiation. The continuous domain theory is presented first. Consider the effect of imposing the following condition on the electric field at z=0:

$$\left(\frac{\partial \tilde{E}}{\partial z} - jk_{z0}\tilde{E}\right)\Big|_{z=0} = 0 \qquad \text{(EQ 46)}$$

Assume the electric field is given by:

$$\tilde{E} = \tilde{a}exp\left(j\left(\omega t + k_x x + k_y y + k_z z\right)\right) + \tilde{b}exp\left(j\left(\omega t + k_x x + k_y y - k_z z\right)\right) \qquad \text{(EQ 47)}$$

where the first term represents the incident field (from z>0, travelling in the -z direction) and the second the reflected field (from the z=0 boundary). Substitute (EQ 47) into (EQ 46) and solve for the reflection coefficient

$$r = \left|\frac{k_z - k_{z0}}{k_z + k_{z0}}\right| = \left|\frac{k_0\cos(\theta) - k_0\cos(\theta_0)}{k_0\cos(\theta) + k_0\cos(\theta_0)}\right| = \left|\frac{\tilde{b}}{\tilde{a}}\right| \qquad \text{(EQ 48)}$$

which is the ratio of the reflected wave amplitude to the incident wave amplitude (plotted vs. angle of incidence in Figure 8.). For incident waves at angles such that $k_z = k_{z0}$

Reflection Coefficient for First Order Boundary Condition

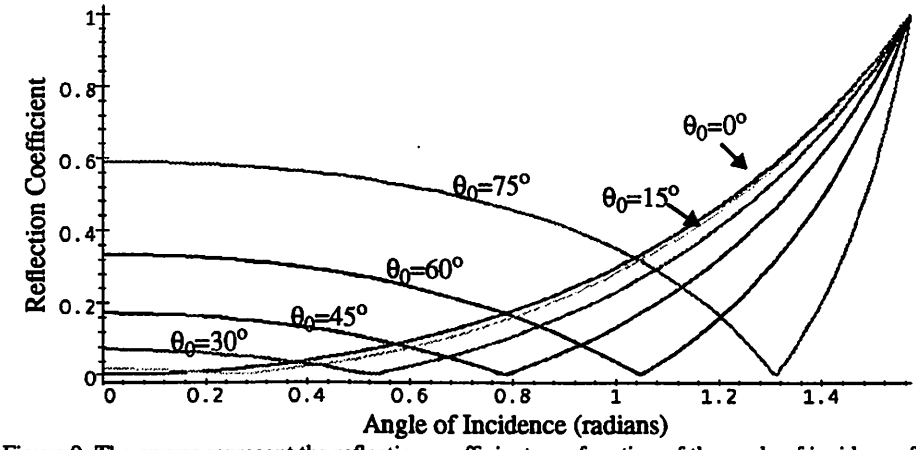

Figure 9. The curves represent the reflection coefficient as a function of the angle of incidence for various θ_0 (angles with no reflection). Note that the reflection coefficient is independent of the azimuthal angle ϕ.

(or $\theta = \theta_0$), there is no reflected wave - the wave is perfectly absorbed. Unfortunately this only happens for one particular angle of incidence. Ideally, all incident waves should be absorbed; but even so, this boundary condition is potentially useful if some reflections can be tolerated. Typically the user of the program would minimize reflections by setting the angle for which there is no reflection to the angle for which the most incidence is expected.

The discretization of (EQ 46) is worked out in [3], and is merely presented here for the case of absorbing waves travelling in the -z direction:

$$\vec{E}^{n+1}[i,j,k] = \vec{E}^{n}[i,j,k+1] + \frac{\omega\Delta t - k_{z0}\Delta x}{\omega\Delta t + k_{z0}\Delta x}\left(\vec{E}^{n+1}[i,j,k+1] - \vec{E}^{n}[i,j,k]\right) \quad \textbf{(EQ 49)}$$

This is the equation used to update the nodes at xy-planes for which the first order boundary condition is used to absorb incident radiation travelling in the -z direction. Similar equations can be written for absorption in other directions. A very important feature of (EQ 49) is that it doesn't require a nearest neighbor from below ($\vec{E}[i,j,k-1]$); for if it did, it would be useless as a boundary condition.

Note also that higher order boundary conditions exist and offer improved angular absorption spectrums (Chapter 4. of [3]). However, it was decided not to program these higher order boundary conditions into TEMPEST 5.0 because they are superceded by the much better PML boundary condition presented in the next section.

4.4 Perfectly Matched Layers (PML) Absorbing Boundary Condition

The Perfectly Matched Layer Boundary Condition ([5],[7],[8],[12]) can be thought of as an imaginary material[1] which has the special property that it absorbs incident radiation, without reflection at all angles of incidence and for all frequencies. Because of this special property, it useful as an absorbing boundary condition.

The continuous theory is only summarized here as it is not necessary to duplicate the existing and quite good explanations in the literature ([5],[7],[8]). Because the discretized equations are quite lengthy (involving up to 30 terms for each type of PML) and not much insight is gained by looking at them, they are not presented in this paper.

4.4.1 PML: Continuous Theory

Consider plane wave reflection from a planar interface between two different materials, A and B (Figure 10.) PML theory can be derived by considering the following (two part) question:

Can one choose $\underline{\varepsilon}_{complex,B}, \underline{\mu}_{complex,B}$ *for material B such that plane waves originating from material A, incident on the infinite planar interface between A and B are*

1. By "imaginary" it is meant that the material cannot be physically constructed.

Set up for the Reflection Problem

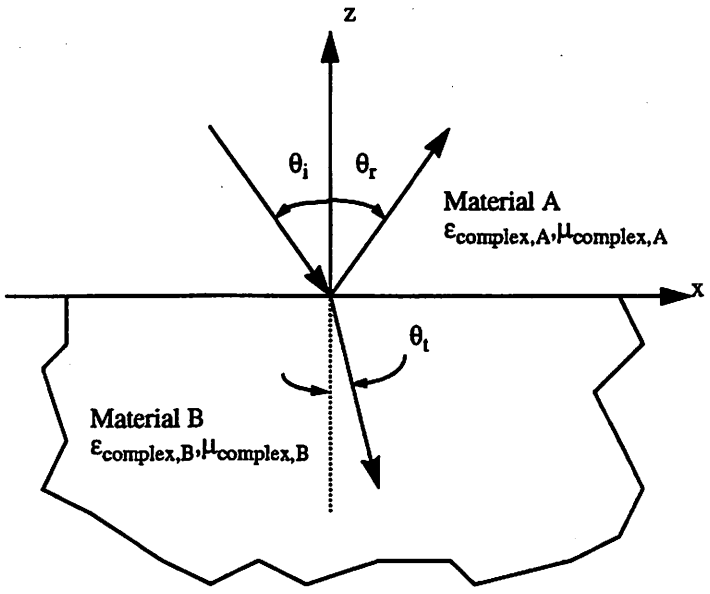

Figure 10. The is the setup for the standard reflection of plane waves problem. The phase matching condition at z=0 will force $\theta_r = \theta_i$ (angle of incidence equals angle of reflection) while other boundary conditions on the electric and magnetic fields will determine the θ_t (direction of transmitted wave) and the magnitude of both the reflected and transmitted waves relative to that of the incident wave. For PML, instead of solving for the reflection coefficient in terms of material properties, the reflection coefficient is set to zero and the material properties are solved for.

not reflected for all angles of incidence θ_i. Furthermore, can one also choose $\underline{\varepsilon}_{complex,B}, \underline{\mu}_{complex,B}$ so that in addition to the zero reflectivity, the transmitted wave is attenuated?

The way of answering this question is to carry out the standard reflection problem [7],[9] (remembering to treat $\underline{\varepsilon}_{complex}$ and $\underline{\mu}_{complex}$ as diagonal tensors), impose the condition that the reflection coefficient be zero and solve for $\underline{\varepsilon}_{complex,B}$ and $\underline{\mu}_{complex,B}$. A zero reflection coefficient is obtained for situations where Material A is isotropic and non-magnetic when[8]:

$$\underline{\varepsilon}_{complex,B}(\omega) = \varepsilon_{complex,A}(\omega) \begin{bmatrix} \kappa + \frac{\sigma}{j\omega\varepsilon_0} & 0 & 0 \\ 0 & \kappa + \frac{\sigma}{j\omega\varepsilon_0} & 0 \\ 0 & 0 & (\kappa + \frac{\sigma}{j\omega\varepsilon_0}) \end{bmatrix}^{-1} \quad \textbf{(EQ 50)}$$

and

$$\underline{\mu}_{complex,B}(\omega) = \mu_{complex,A}(\omega)\begin{bmatrix} \kappa + \frac{\sigma}{j\omega\varepsilon_0} & 0 & 0 \\ 0 & \kappa + \frac{\sigma}{j\omega\varepsilon_0} & 0 \\ 0 & 0 & (\kappa + \frac{\sigma}{j\omega\varepsilon_0})^{-1} \end{bmatrix} \quad \text{(EQ 51)}$$

where

$$\varepsilon_{complex,A}(\omega) = \varepsilon_0\left(\varepsilon_r + \frac{\sigma_e}{j\omega\varepsilon_0}\right)$$
$$\mu_{complex,A}(\omega) = \mu_0\mu_r \quad \text{(EQ 52)}$$

are the material constants for Material A.

The attenuation of the transmitted wave in Material B is controlled by the κ and σ parameters[8].

4.4.2 Using PML as a Boundary Condition in the Simulation Domain

The usefulness of this PML material as a boundary condition can be understood with the aid of an example. Suppose one wants to simulate a point source excitation in free space. The point source is positioned in the center of the simulation domain and the domain is surrounded with PML set to absorb in the outgoing directions. This simulation is illustrated in Figure 11.

Highlighted in the figure are three types of PML called 1D, 2D and 3D PML. As can be seen, 1D PML is used on the faces of the rectangular domain, 2D PML is used at the edges of the domain where two 1D PML slabs would intersect, and 3D PML is used in the corners where three 1D PML slabs would intersect. As expected, 1D PML attenuates in one direction, 2D PML attenuates in two directions, and 3D PML attenuates in three directions. Each block of the PML material differs from the others in the direction(s) of attenuation and thus the material constants, $\underline{\varepsilon}_{complex}$ and $\underline{\mu}_{complex}$, for each block are different.

As the outgoing waves approach the PML material, they pass into the material without reflection and are attenuated. The degree of attenuation in the PML is dependent on κ and σ and also the thickness of the PML. If the degree of attenuation is large enough, the amount of energy reaching the true edges of the domain is insignificant and thus any of the above boundary conditions could be used since any energy reflected back into the domain would be insignificant.

Placing PML around the edges of the Simulation Domain

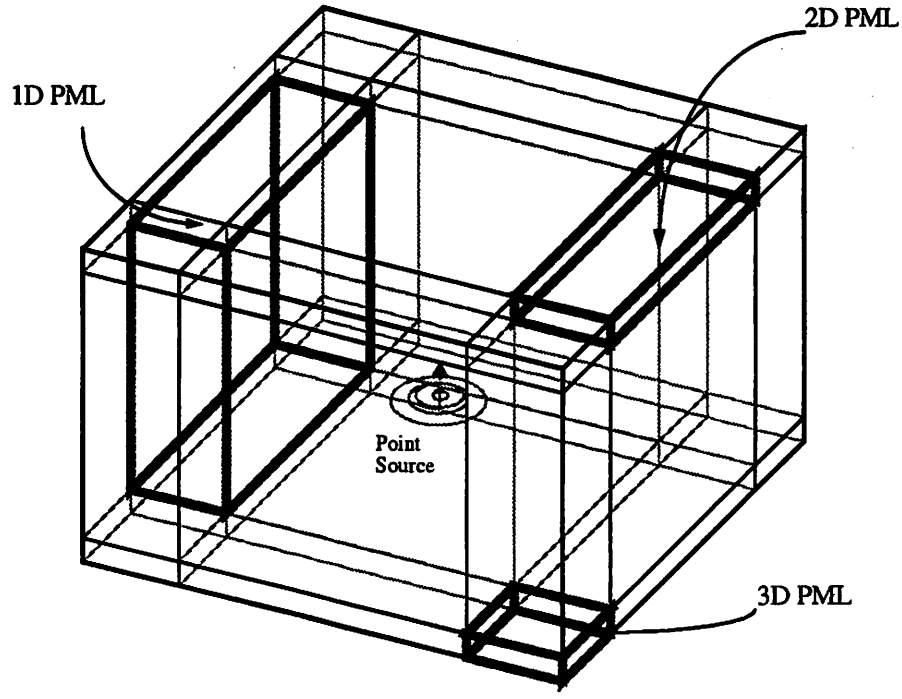

Figure 11. The PML material is placed around the domain. Ideally, all outgoing radiation (from the point source at the center of the box) is absorbed by the PML without reflection. Effectively the solution for the radiating point source in free space is found for the finite region of space modelled by the domain. Note that PML material is different (in its direction(s) of attenuation) for each face, edge and corner of the domain.

4.4.3 The Discretized Equations

The discretization of the Maxwell curl equations with (EQ 51) and (EQ 52) follows essentially the procedure outlined in Chapter 2 - especially the second example in Section 2.2.3. For exact details of the discretization of the PML material equations the reader is referred to [8].

4.4.3.1 Variation of the Material Parameters from Node to Node

The continuous PML theory works in the continuous domain but not in the discretized domain. In [5] it is mentioned that an abrupt change in $\underline{\varepsilon}_{complex}$ or $\underline{\mu}_{complex}$ in a finite difference scheme will produce reflections. For this reason, the κ and σ parameters (which provide the attenuation) must start at 1 and 0 respectively at the interface between the PML and the adjacent material, and then be slowly "ramped up" from node to node as one goes deeper into the PML material to provide the attenuation. In [10] it was determined that a quartic variation in these parameters was optimal. 1D PML for example, attenuates in only one direction so the κ and σ parameters vary from node to node in only one direction. The 1D PML material thus looks like a stack of "layers" each "perfectly matched" to its neighboring layers so as not to produce reflections - hence the name "Perfectly Matched Layers".

4.4.3.2 Dealing with the Staggered Grid

Another issue of concern is the staggered grid (see Chapter 2). Different field components sit at different positions within a node. Inside the PML, in the attenuating directions, the variation of κ and σ within the node must be considered meaning that the updating equations for different field components within a node are based on different values of κ and σ. To calculate the value of κ and σ used for any particular field component, TEMPEST averages the continuous quartic variation over the length Δx centered around the field component's location.

4.4.3.3 Computational Intensity

The variation of the material parameters presents another difficulty. Take for example a block of 3D PML say 8x8x8 nodes. Since the PML absorbs in three directions, the parameters κ and σ must vary from node to node in all three directions yielding $8^3 = 512$ different updating equations - one for each node. To store the coefficients for each of the 512 updating equations requires a lot of memory so instead, TEMPEST retains the dimensions of the block of PML and calculates for each node, knowing the position within the block of PML, the κ and σ parameters and the updating coefficients *each time the node is updated.*

The computational intensity of PML has been summarized in Table 1 of Chapter 2 and can be compared to those for other materials. Not accounted for in the table is the computation necessary to calculate the updating coefficients as described in the previous paragraph. Again note the usefulness of being able to solve more general equations for only the nodes in the domain which require them.

4.5 TEMPEST 5.0 Implementation of Boundary Conditions

The perfect conductor, first order, and PML boundary conditions are considered as "special materials" by TEMPEST 5.0. They occupy nodes in the simulation domain and have associated updating equations. For the perfect conductor boundary condition the updating equation is to set the fields to zero. For the PML boundary condition the updating equations are the discretized Maxwell equations for an anisotropic magnetic material. In the case of the first order boundary condition, the updating equations are ones that involve only five out of the six nearest neighbors. There is no reason why these boundary condition nodes must be at the edges of the domain. They could be just as well put anywhere inside the domain and TEMPEST 5.0 allows the user to easily do this.

The periodic and symmetric boundary conditions do not occupy nodes as do the other boundary conditions. Instead they "connect" nodes at the edges of the domain to other nodes within the domain.

CHAPTER 5

Example Simulations

An ongoing goal in the development of TEMPEST is to be able to simulate ever larger problems. There are several factors involved in achieving this goal. The most notable are the continuous improvements in computer technology and the continuous development of electromagnetic simulation techniques such as the PML boundary condition which is used to truncate the domain. The first section of this chapter presents some simulation results which demonstrate the effectiveness of the PML and its ability to truncate a domain while the second section presents a large reflective notching simulation which employs PML and serves to demonstrate the current overall capability of the TEMPEST program.

5.1 PML Demonstrations

A thorough characterization of the PML boundary condition's performance is a hefty task. One must measure reflection coefficients of plane waves at many different angles of incidence, many frequencies, with various thicknesses of PML and various variations of the PML material properties. Other issues such as nonhomogeneous materials adjacent to the PML and evanescent field attenuation further compound the problem. No attempt is made here to quantify the PML. Only qualitative results are presented.

5.1.1 1D PML

One dimensional PML (PML which attenuates in only one direction) is often used in simulations where only the top and bottom faces of the domain are to be isolated while the remaining side faces employ periodic or symmetric boundary conditions. This allows the simulation of infinite planar surfaces and interfaces. The following figure

(Figure 12.), shows three simulations which demonstrate how PML absorbs a single plane wave at three different angles of incidence.

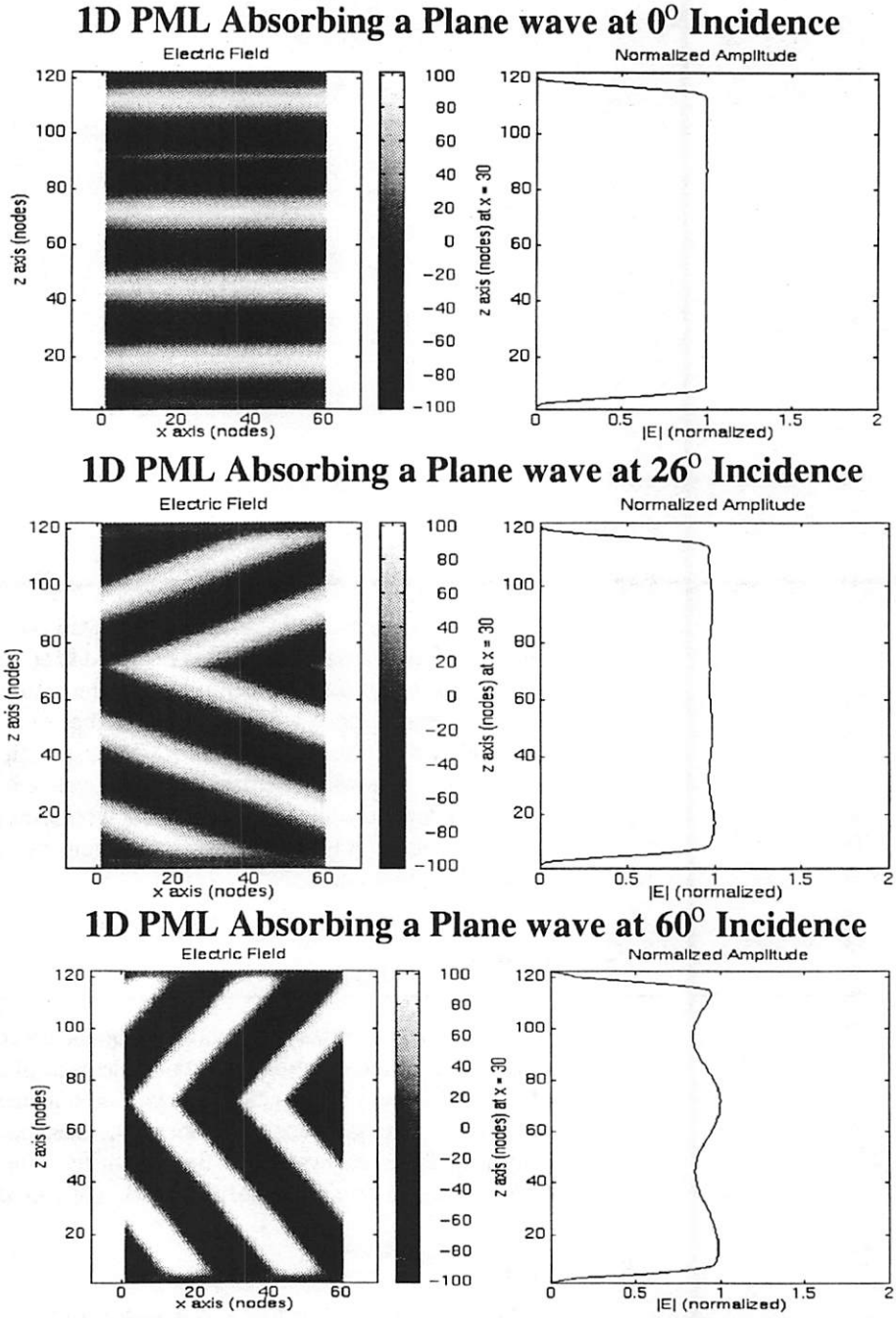

Figure 12. The left hand plots are the electric fields in steady state. The right hand plots are the time averaged electric field amplitudes at the x=30 node. Note how the standing wave ratio increases with the angle of incidence implying more reflections for higher angles of incidence. The "kink" in the fields is centered on the plane of excitation and is due to the bi-directional propagation of energy from the plane source.

The top diagrams in Figure 12. demonstrate absorption of a normally incident (z - directed) plane wave. The left hand side diagram shows the instantaneous steady state electric field in the zx plane and the right diagram shows a vertical cutline of the electric field amplitude at the node x = 30 (right up the middle of the left diagram). The almost perfectly straight vertical line in the right diagram shows that no standing wave has developed and also shows how the field decays within the PML at the top and the bottom.

The diagrams in the middle and at the bottom show cases for plane waves approaching at oblique angles. Note that a standing wave pattern begins to develop at higher angles of incidence indicating that the PML is not perfectly reflectionless.

5.1.2 2D PML and 3D PML

The following 2D simulation (Figure 13.). demonstrates the effectiveness of 2D PML (in the edge regions) and the use of planar segment excitation.

Figure 14. is a 3D simulation of a point source in free space. The PML absorbing boundary condition is used to truncate the domain. This demonstrates 1D, 2D and 3D PML. The topography (with the boundary conditions included) is similar to that illustrated in Figure 11., Chapter4 and the cutplane where the electric field is being observed is an zx cutplane passing through the excited node.

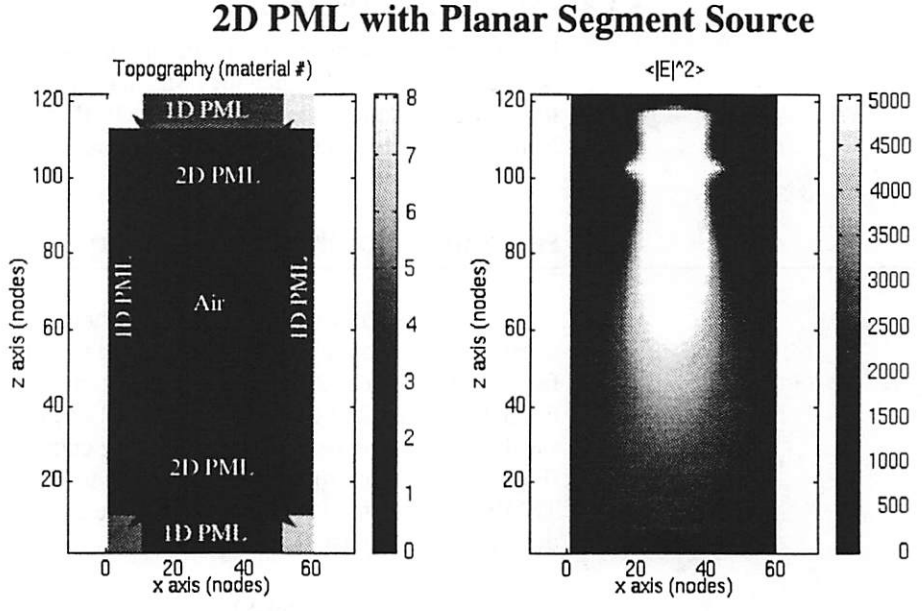

Figure 13. The left hand figure shows the topography of the simulation - free space with PML at the edges of the domain. Note that the corner regions are 2D PML. The right hand plot is the intensity. Note that a planar segment excitation was used. Note how there are no visible reflections from the domain edges - PML appears to be working well.

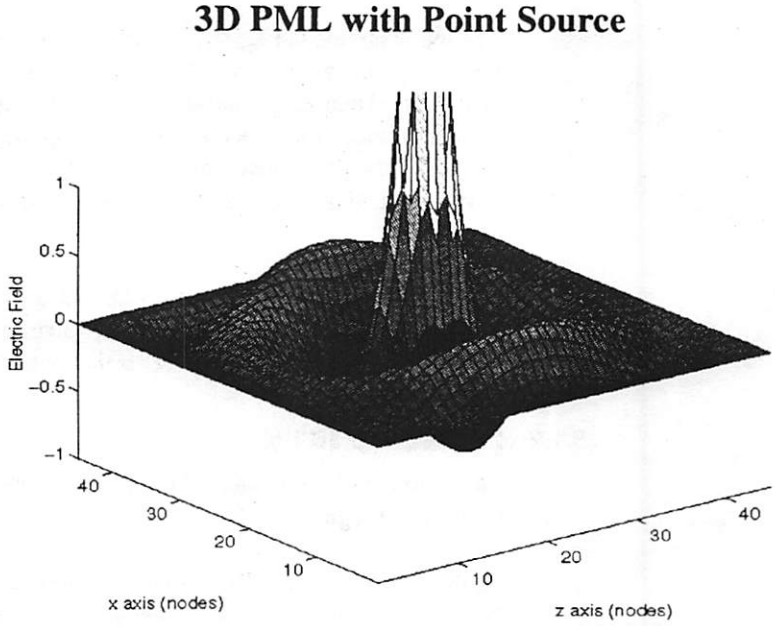

Figure 14. The electric field from a z-directed point source. Note how the ripples in the surface just "die out" at the edges of the domain where the PML is.

5.1.3 PML at Material Interface

Often, two or more materials are present on a single face of the domain. If this face is to be in contact with PML, then the PML must match whichever material it is next to. The following simulation (Figure 15.) shows PML matched across an air-resist interface. Note that no reflections from any of the domain edges are apparent.

5.2 Reflective Notching Simulation

The final demonstration is a reflective notching simulation. The topography (Figure 16.) consists of a reflective nickel topography at the bottom, resist with a non-horizontal surface angle, and air at the top. The illumination is a rectangular image projected down through the resist and down onto the nickel surface. TEMPEST was asked to output the steady state electric field intensity in the yz cutplane at the side wall of the illumination (Figure 17.). The uniformity of the electric field intensity inside the resist is of concern. TEMPEST was used to see how closely the exposure of the resist matches the projection of the rectangular object

This simulation demonstrates many of the features of TEMPEST 5.0. First and foremost, TEMPEST's ability to solve large problems in a reasonable amount of time is demonstrated. The domain size is 76 by 334 by 200 = 5.0768 million nodes. The TEMPEST 5.0 algorithm requires approximately 150Mbytes of memory and 6 hours on an Ultra SPARC 2, 200MHz workstation. Secondly the PML is used at the top and at the y borders to absorb reflected light heading out of the domain and symmetric boundary conditions are employed in the x direction effectively making the simulated topography

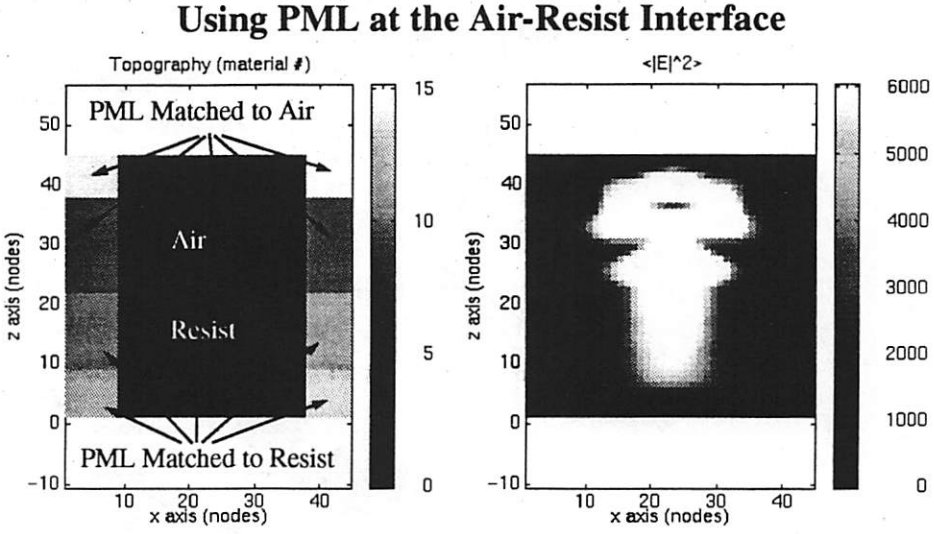

Figure 15. The left hand figure shows the topography. The top half of the domain is air and the bottom half of the domain is resist. The entire domain is surrounded by PML - 1D PML at the faces and 2D PML in the corners. Note that the PML in contact with the air is matched to the air and the PML in contact with the resist is matched to the resist. From the figure on the right (the intensity), it appears that this scheme doesn't seem to pose a problem.

a symmetric object. The generalized illumination allows the projection of a defocused image at the top of the domain which propagates in the downward direction coming to a focus somewhere inside the resist.

For further details of this particular simulation see [14].

Reflective Notching Simulation Topography

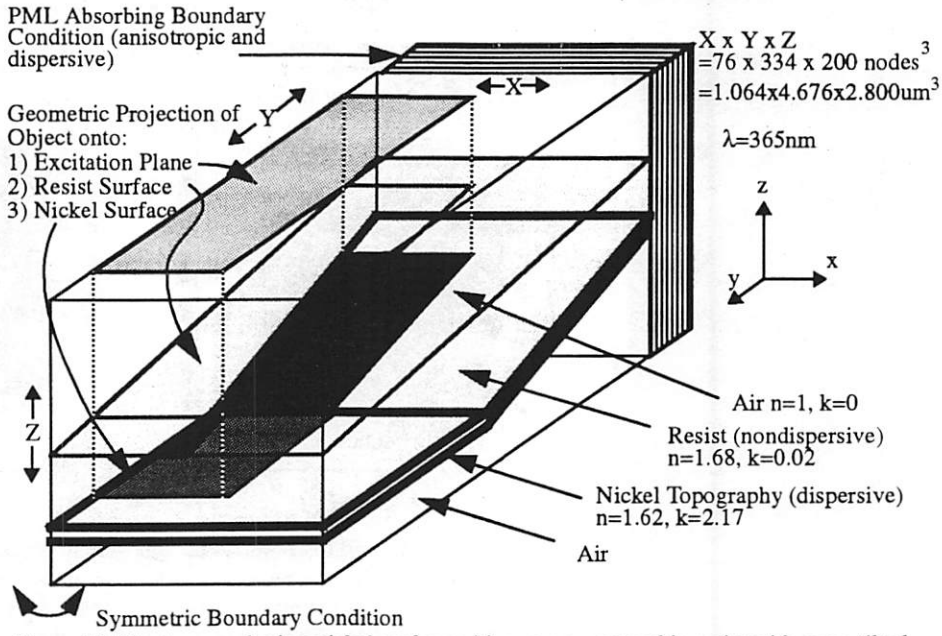

Figure 16. The topography is a nickel surface with a corner covered in resist with some tilted surface angle. The illumination is rectangular and exposes the region of the resist which will be the trench. The uniformity of the illumination in the trench side walls (plane with normal in the x direction) is of interest.

Simulation Result: The Electric Field Intensity in the yz Plane

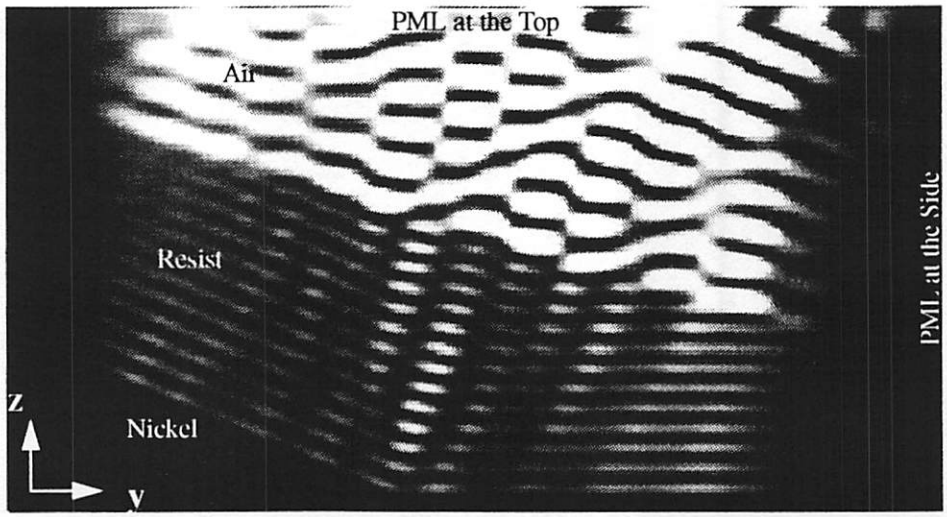

Figure 17. This is the electric field intensity in the yz plane at the edge of the illumination. Ideally the resist should be uniformly illuminated but because of the reflective nickel underneath, standing wave patterns are created. Also complicating the matter are the non-horizontal resist surface angle and the angled nickel section. Some "streaks" or "ripples" seem to be emanating from the corner.

CHAPTER 6

Summary

6.1 Overall Program Architecture

The most important improvement to TEMPEST is its new ability to update different nodes with different updating equations. This enables TEMPEST to simulate a much broader range of materials (anisotropic, PML, 1st order BC's, dispersive, magnetic). This is also efficient since more computation time and memory are devoted to only those nodes which require them.

TEMPEST 5.0 allows the user to have multiple plane and point sources in the simulation. This gives the user much more flexibility and control over the illumination and opens up a whole new range of simulation possibilities. Convergence checking has been improved by considering the convergence of the fields throughout the entire domain (rather than in a single plane of nodes) making it less likely for a false convergence to occur.

A new and promising boundary condition, Perfectly Matched Layers, has been added to TEMPEST. The boundary condition, along with the basic first order boundary condition can now be placed anywhere in the domain allowing the simulation of isolated topographies - something not possible in TEMPEST 4.0.

6.2 Other Improvements to TEMPEST

There are many other "spin-off" improvements to TEMPEST not detailed in this report but worth mentioning:

- Memory Savings - TEMPEST 5.0 uses much less memory than TEMPEST 4.0 for two reasons. The first and most significant reason is that TEMPEST 5.0 uses a different memory structure than TEMPEST 4.0. The second is because for a given

material type, version 5.0 stores only the necessary field components whereas version 4.0 stores the same number of field components for each node since all nodes are updated with updating equations of the same generality. A typical simulation with TEMPEST 5.0 requires approximately 30 bytes/node - approximately a factor of three better than version 4.0.

- Binary Output Format - TEMPEST 5.0 allows the outputs to be written to file in the binary format which take up considerably less space than the version 4.0's output format and are easily imported into software such as MATLAB.
- Other minor improvements:

  -fixed bug in definitions of Δx and Δt,

  -removed unnecessary iterations while writing output files,

  -refractive index plots are now written to disk before the simulation starts enabling the user to verify correctness of topography without having to run the entire simulation.

6.3 The Future of TEMPEST

One of the main advantages of the Finite Difference Time Domain method is its ease of parallelizability. Originally, in 1989, TEMPEST was written for a parallel machine [1], but since then the need for a workstation version was realized and TEMPEST 4.0 was written for a workstation. Now the increasingly popular trend is parallel processing over a network of workstations and for TEMPEST this has tremendous potential. Currently a prototype parallel version of TEMPEST for workstations is being developed.

TEMPEST 5.0 employs a constant nodal density gridding of the domain. A significant memory savings may be achieved by allowing multiple nodal densities. As a general rule, 15 nodes per wavelength are required in the most optically dense medium which may result in significantly more nodes than necessary in media which are not optically dense such as air. This waste of nodes might be alleviated if it were possible to assign different nodal densities to the air and to the optically dense materials.

TEMPEST 5.0 is also an excellent tool for the further investigation of PML. How do isolated sections of PML behave? What about PML for lossy, anisotropic material? What happens when PML is used at material interfaces? The answers to these any many more questions concerning PML might be found by simulation with TEMPEST.

Currently research is being conducted to allow TEMPEST to simulate partial coherence effects [11], surface roughness [15] and multilayer thin film structures.

References

[1] Gamelin, J., "Simulation of Topography Scattering for Optical Lithography with the Connection Machine", M. S. Thesis, Memorandum No. UCB/ERL M89/71, University of California, Berkeley, May 1989.

[2] Guerrieri, R., Tadros, K. H., Gamelin, J., Neureuther, A., "Massively Parallel Algorithms for Scattering in Optical Lithography," *IEEE Trans. CAD*, vol. 10, no.9, pp.1091-1100, Sept. 1991.

[3] Wong, A., *Rigorous Three-Dimensional Time-Domain Finite-Difference Electromagnetic Simulation,* Electronics Research Laboratory, University of California, Berkeley, 1994

[4] Yee, K. S., "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," *IEEE Trans. Antennas Propagation*, vol. 14, pp. 302-307, May 1966.

[5] Berenger, J., "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves", *Journal of Computational Physics*, 114, pp185-200, 1994.

[6] Kunz, K. S., Luebbers, R. J., *The Finite Differnece Time Domain Method for Electromagnetics*, CRC Press, 1993.

[7] Cangellaris, A., Zhao, L., *GT-PML: Generalized Theory of Perfectly Matched Layers and Its application to the Reflectionless Truncation of Finite-Difference Time-Domain Grids,* Electromagnetics Laboratory, Department of Electrical and Computer Engineering, University of Arizona, 1996.

[8] Gedney, S. D., "An anisotropic PML Absorbing Media For the FDTD Simulation of Fields in Lossy and Dispersive Media", *Electromagnetics*, vol. 16, No. 14, pp. 399-415, July-August, 1996.

[9] Jackson, J. D., *Classical Electrodynamics*, John Wiley & Sons, 1975.

[10] Gedney, S., "An Anisotropic Perfectly Matched Layer Absorbing Media for the Truncation of FDTD Lattices," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 12, December, 1996.

[11] Socha, R. J., Neureuther, A. R., "Propagation effects of partial coherence in optical lithography", J. Vac. Sci. Technol. B 14(6), Nov/Dec 1996.

[12] Katz., D, Thiele, E., Taflove, A., "Validation and Extension to Three Dimensions of the Berenger PML Absorbing Boundary Condition for FD-TD Meshes," *IEEE Microwave and Guided Wave Letters*, Vol. 4, No. 8, August 1994.

[13] Luebbers, R., Hunsberger, F., Kunz, K., Standler, R., Schneider, M., "A Frequency-Dependent Finite-Difference Time-Domain Formulation for Dispersive Materials," *IEEE Transactions on Electromagnetic Compatibility*, Vol. 32, No. 3, August, 1990.

[14] Pistor, T., Neureuther, A. R., "Simulation of Reflective Notching with TEMPEST", Optical/Laser Microlithography X, SPIE, 1997.

[15] Adam, K., Socha, R., Pistor, T., Neureuther, A., "Simulating Photmask Edge Roughness and Corner Rounding", Optical/Laser Microlithography X, SPIE, 1997.