National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

A New Finite-Difference Time-Domain Method Applied to an Open Waveguide Structure

by
Michael Fitzmaurice, B.A. Sc.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering

Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

ISBN 0-315-85755-2

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

Abstract

This study makes use of a variation of the Finite-Difference Time-Domain (FDTD) method as first proposed by Yee to simulate electromagnetic field distribution and propagation in an open waveguide structure. In order to prove that this new method is valid, a reflection coefficient is calculated with simulation data and compared to measurements. The agreement between measurement and simulation data, while not exact, is enough to establish the veracity of the new method. This study contains a detailed discussion of the discrepancies which were observed. Also presented are colour images of the simulation which give the reader an idea as to the nature and level of detail of the information which can be obtained from the simulation.

Acknowledgements

The author would like to acknowledge the support and encouragement of Dr. M. Ney and the many helpful discussions with fellow graduate students. In particular, the author would like to thank Z. Chen for his advice and patience without which this study could not have been completed.

The author would also like to acknowledge the loving support and editorial prowess of his wife Tracy; she rightly shares in any of his accolades.

Contents

List of Figures

Chapter 1

Background

1.1 Introduction

Computer simulations of electromagnetic (EM) fields are becoming more widespread with each passing year. This trend is due primarily to the inexorable advances in computer technologies resulting in more people having access to increasingly powerful computers. In [2], Taflove remarked on this trend warning government agencies that computer simulations of complex EM phenomena will no longer be the exclusive domain of western nations and their research organizations.

In the past, computer simulations of microwave structures were not feasible; computers ran too slowly and their capability for graphical presentation was extremely limited. Today, the response of simple microwave structures can be simulated interactively and displayed using colour and graphical images. Computer memory and processing speed are still a limiting factor; the more complicated the structure being simulated, the more computer "horsepower" (i.e. speed and memory) is required to achieve results in a reasonable length of time. Simulations of open structures are especially onerous on computers due to the sheer size of the problem space that is usually desired to model the free space environment.

Only the powerful, large and expensive mainframe computers (i.e. CRAY) are capable of making short work of simulations involving more complicated microwave structures such as waveguide and microstrip discontinuities, junctions and open structures. Unfortunately, access to these types of machines is restricted and costly.

1.2 Literature Survey

Yee's work [1] is widely considered to be the seminal paper regarding the FDTD method. The idea of using finite difference equations as an approximation to differential equations is not new or restricted to electrical engineering applications (as noted in [3]). However, before Yee's time, non-analytic solutions to "real" EM problems often involved solving numerous equations over hundreds of iterations. This task was too demanding for the contemporary computers and beyond the patience of most humans. As a result, there was little interest in computer simulations of anything but the simplest microwave scatters and structures (i.e. perfectly conducting spheres, cylinders, resonant cubic cavities etc.).

Yee, in a sense, showed the EM community that computers had come of age and could begin to deal with Maxwell's formidable equations. His work involved the two-dimensional simulation of scattering caused by a perfectly conducting cylinder and wedge based upon what later became known as the FDTD method.

A comprehensive history of the development of the FDTD method is contained in [4] and will not be repeated here. This chapter will concentrate more upon the various applications of the method by others in an effort to show its adaptability and growing acceptance within the EM and numerical modelling communities.

The effects of Electromagnetic Pulses (EMPs) upon the human body and metallic structures and scatterers have always been an important area of research. The FDTD method has been used extensively in this field to predict EMP effects. Holland [5] developed a computer program, known as THREDE, which could calculate free-field EMP coupling and scattering in a $30 \times 30 \times 30$ problem space. Kunz and Lee [6] made use of this program to model EMP effects upon a non-operational F-111 aircraft and then compared predicted results to actual measurements. The agreement between the two was very good.

Holland *et al.* [7] expanded THREDE to a $50 \times 43 \times 59$ problem space and used it to examine EMP scattering by a lossy dielectric sphere. The data generated by THREDE compared favourably to the Rayleigh-Mie spherical harmonic expansion solution. Holland and Simpson [8] altered THREDE again so that it could accommodate thin wires and struts within the problem space. The authors then went on to investigate EMP coupling into a wire loop and dipole antennas and found their data to be very close to the known solutions. Chen and Gandhi [9] used the FDTD method

to predict internal fields and currents induced in the human body by an EMP. Like the others, Chen and Gandhi's predictions agreed favorably with measured data.

The effects of EM radiation on human tissue is another important area of research. In the treatment of cancer patients it is desirable to restrict radiation, as much as possible, to the affected tissue (i.e. tumor). This entails accurate knowledge of specific absorption rates (SAR) for various tissues and near field radiation characteristics for radio frequency (RF) applicators. Sullivan [10] has used the FDTD method to predict the SAR distribution produced by an applicator with a view to improving its design. Sullivan *et al.* [11] employed the FDTD method to calculate EM absorption in human tissue. In particular, the authors were concerned with determining the local SARs at specific points within the human body and their work confirmed the suitability of the method for this type of research. Wang and Gandhi [12] used the method to simulate performance of a device used for treatment of cancer patients: the Annular Phased Array (APA). These authors suggested that the method was well suited for near field simulations of realistic RF applicators used for hypothermia and other medical applications.

Microwave circuit designers have also seized upon the FDTD method as an aid to building better circuits and transmission line structures. Moore and Ling [13] used the method to optimize the length of a 45° miter in a 90° bend in microstrip. Wolff and Rittweger [14] used the FDTD method to examine the responses of an open ended line, an impedance step, a meander line, a radial stub and a spiral inductor all in microstrip. Their results compared favorably to measured data. Their analysis of the spiral inductor is particularly encouraging because this component is very complicated; it includes an air bridge above the surface of the substrate material. Zhang and Mei [15] examined the characteristics of six common microstrip structures using FDTD and succeeded in establishing the validity of the method for obtaining frequency domain results.

The FDTD method has also found use in the analysis of EM field scattering and radar cross section (RCS) determination. Taflove and Umashankar [3] used the method to examine simple geometrical 2D and 3D targets. Their results compared favorably with measurements and MOM (Method of Moments) data. The authors also examined two particularly complex structures: a missile seeker section and the human body. The analysis upon these two structures demonstrates the extremely detailed information that the method can provide and the complexity of the object

that can be simulated. Umashankar and Taflove combined their efforts again [16] and showed how FDTD data can be combined with other procedures to yield the far field scattering of complex objects. FDTD was used to determine the near field of a scatterer. This data was then transformed to the far-field scattering pattern based on EM equivalences. The RCS of the complex scatterer could then be determined.

The FDTD method proved to be well suited to all the aforementioned applications. A common concern expressed was that of memory storage and CPU utilization time (that concern will be echoed in this study as well). Although the FDTD method has some advantages over other methods (for example TLM as discussed in [17, 18]), the primary restriction is still the amount of computer resources that can be allocated to the task.

Next to [1], the paper produced by Chen *et al.* [17] is of primary importance as far as this study is concerned. Chen outlined a new FDTD scheme and showed how it was equivalent to the symmetrical condensed TLM node. Chen's new scheme provides better convergence at the expense of increased computations (i.e. more numerical operations). In the Yee cell there are eighteen field components while Chen's new cell uses 30. More field components provide for better resolution yet Chen observed that the time required for the computations was approximately the same for both methods.

1.3 Motivation

In general, the purpose of this study is to further the acceptance of the FDTD method as a valid and useful tool for accurately simulating the performance of microwave devices. In particular, this study aims to show the practical application of the FDTD method, as modified by Chen, to a specific waveguide structure.

Henceforth, the terms "the FDTD method" or "the method" will refer to the FDTD method, originally proposed by Yee [1] but later modified by Chen *et al* [17]. In order to achieve the purpose of this study, the author will explain the method and demonstrate its use on a specific waveguide structure.

The author is an officer in the Canadian Navy who has been seconded to the University of Ottawa in order to complete a post-graduate degree. The selection of the structure for simulation in this study was based upon two considerations: First, to benefit the Navy, and secondly, to show that this method can satisfactorily deal with

non-canonical structures. The structure simulated is an element of an experimental naval radar array known as EXPAR (Experimental Phased Array Radar).

The Navy is considering further development of EXPAR with a view to procurement; therefore, it is hoped this study will aid in some way in evaluating the technical merits of the system.

Chapter 2

Theory

2.1 Introduction

Yee's FDTD method has been examined and explained by himself and numerous other authors [1, 3, 15, 14, 19, 18, 10]. Chen's modification of the FDTD scheme (which is used in this study), although fully developed in [17], was not presented in its entirety probably due to brevity constraints. This chapter will, therefore, completely develop and present all equations required by the FDTD method.

The equations necessary for three-dimensional simulations will be developed from Maxwell's two curl equations in a stationary and sourceless medium. The boundaries used in the FDTD problem space will then be presented followed by a discussion of the stability aspects of the simulation. Finally, a simple cavity structure will be simulated using the FDTD method and the results will be compared to its analytical solution.

2.2 The FDTD Method

Maxwell's curl equations in a stationary and sourceless medium are:

$$\mu \frac{\partial \vec{\mathrm{H}}}{\partial t} = -\vec{\nabla} \times \vec{\mathrm{E}} \tag{2.1}$$

$$\epsilon \frac{\partial \vec{\mathrm{E}}}{\partial t} = \vec{\nabla} \times \vec{\mathrm{H}} \tag{2.2}$$

where

$$\vec{\mathrm{E}} = E_x \hat{x} + E_y \hat{y} + E_z \hat{z}$$

$$\vec{\mathrm{H}} = H_x \hat{x} + H_y \hat{y} + H_z \hat{z}$$

in Cartesian coordinates.

All vector components are functions of space (x,y,z) and time (t). For example, $E_x = E_x(x, y, z, t)$. Equations 2.1 and 2.2 can be expanded as follows:

$$\mu\left[\frac{\partial H_x}{\partial t}\hat{x} + \frac{\partial H_y}{\partial t}\hat{y} + \frac{\partial H_z}{\partial t}\hat{z}\right] = \tag{2.3}$$
$$-\left[\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z}\right]\hat{x} - \left[\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x}\right]\hat{y} - \left[\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y}\right]\hat{z}$$

$$\epsilon\left[\frac{\partial E_x}{\partial t}\hat{x} + \frac{\partial E_y}{\partial t}\hat{y} + \frac{\partial E_z}{\partial t}\hat{z}\right] = \tag{2.4}$$
$$\left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z}\right]\hat{x} + \left[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x}\right]\hat{y} + \left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right]\hat{z}$$

Equating vector components in 2.3 and 2.4 yields:

$$\mu\frac{\partial H_x}{\partial t} = \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \tag{2.5}$$
$$\mu\frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \tag{2.6}$$
$$\mu\frac{\partial H_z}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \tag{2.7}$$
$$\epsilon\frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \tag{2.8}$$
$$\epsilon\frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \tag{2.9}$$
$$\epsilon\frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \tag{2.10}$$

As was done in [1] and [17], assume a three-dimensional, discretized Cartesian grid which will henceforth be referred to as the problem space. The intersection of the grid lines represents a point or "node" in the problem space. Any node in the space is defined as $(i, j, k) = (i\Delta x, j\Delta y, k\Delta z)$ where Δx,Δy, and Δz are elemental distances in the $\hat{x}$,$\hat{y}$, and $\hat{z}$ directions. Any function of space and time can be expressed as:

$$F(i\Delta x, j\Delta y, k\Delta z, n\Delta t) =_n F(i, j, k)$$

Where i,j,k,n are all integer variables and Δt is the incremental amount by which time is advanced in the simulation.

For this study $\Delta x = \Delta y = \Delta z \equiv \Delta$.

In [20, page 475, eq 13.2.5a], Haberman uses the Taylor series to develop an expression for the first derivative of a function, $f(\omega)$:

$$\frac{df(\omega)}{d\omega} = \frac{f(\omega + \Delta\omega) - f(\omega)}{\Delta\omega} - \frac{\Delta\omega}{2}\frac{d^2 f(\xi_2)}{d\omega^2}$$

The last term represents the Taylor series remainder or truncation error. A finite difference approximation to $df(\omega)/d\omega$ is known as the forward difference approximation:

$$\frac{df(\omega)}{d\omega} \approx \frac{f(\omega + \Delta\omega) - f(\omega)}{\Delta\omega}$$

The forward difference approximation becomes increasingly accurate with decreasing $\Delta\omega$. Haberman also defines a centered difference approximation of $df(\omega)/d\omega$ as:

$$\frac{df(\omega)}{d\omega} \approx \frac{f(\omega + \Delta\omega) - f(\omega - \Delta\omega)}{2\Delta\omega} \tag{2.11}$$

If $\Delta\omega = \Delta\omega/2$, equation 2.11 becomes:

$$\frac{df(\omega)}{d\omega} \approx \frac{f(\omega + \Delta\omega/2) - f(\omega - \Delta\omega/2)}{\Delta\omega} \tag{2.12}$$

Consider equation 2.5. Let $n + \frac{1}{2}$ and (i, j, k) be the centre points in time and space respectively. If the centered difference approximation, as shown in equation 2.12, is applied to the left and right hand side of equation 2.5, the following is obtained:

$$\mu\left[\frac{{}_{n+1}H_x(i,j,k) - {}_nH_x(i,j,k)}{\Delta t}\right] = \tag{2.13}$$
$$\frac{{}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2}) - {}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2})}{\Delta z} - \frac{{}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) - {}_{n+\frac{1}{2}}E_z(i,j-\frac{1}{2},k)}{\Delta y}$$

Expressing 2.13 in terms of ${}_{n+1}H_x(i,j,k)$ and recalling that $\Delta = \Delta z = \Delta y$, the following is obtained:

$$\begin{aligned} {}_{n+1}H_x(i,j,k) \;=\; & {}_nH_x(i,j,k) + \frac{\Delta t}{\mu\Delta}\Big[{}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2}) - \\ & {}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2}) - {}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}E_z(i,j-\frac{1}{2},k)\Big] \end{aligned} \tag{2.14}$$

For the purposes of this study, let Δ=1 mm and Δt=3.3333$\times 10^{-12}$ seconds (the reason for this choice of Δ and Δt will be discussed later). Therefore,

$$\frac{\Delta}{\Delta t} = c_o = \frac{1}{\sqrt{\mu_o \epsilon_o}} = 3 \times 10^8\, m/sec$$

If both sides of 2.14 are multiplied by Z_o where $Z_o = \sqrt{\frac{\mu_o}{\epsilon_o}}$, then,

$$\begin{aligned}\frac{Z_o \Delta t}{\mu \Delta} &= \sqrt{\frac{\mu_o}{\epsilon_o}} \frac{1}{c_o} \frac{1}{\mu} \\ &= \sqrt{\frac{\mu_o}{\epsilon_o}} \sqrt{\mu_o \epsilon_o} \frac{1}{\mu_o \mu_r(i,j,k)} \\ &= \frac{1}{\mu_r(i,j,k)}\end{aligned}$$

$\mu_r(i,j,k)$ refers to the value of the relative permeability at the specific location or node (i,j,k). Equation 2.14 can now be rewritten as:

$$\begin{aligned}{}_{n+1}H_x(i,j,k)Z_o &= {}_nH_x(i,j,k)Z_o + \frac{1}{\mu_r(i,j,k)}\Big[{}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2}) - \\ &\quad {}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2}) - {}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}E_z(i,j-\frac{1}{2},k)\Big]\end{aligned} \tag{2.15}$$

Through similar manipulations, equations 2.6 to 2.10 are transformed to the following:

$$\begin{aligned}{}_{n+1}H_y(i,j,k)Z_o &= {}_nH_y(i,j,k)Z_o + \frac{1}{\mu_r(i,j,k)}\Big[{}_{n+\frac{1}{2}}E_z(i+\frac{1}{2},j,k) - \\ &\quad {}_{n+\frac{1}{2}}E_z(i-\frac{1}{2},j,k) - {}_{n+\frac{1}{2}}E_x(i,j,k+\frac{1}{2}) + {}_{n+\frac{1}{2}}E_x(i,j,k-\frac{1}{2})\Big]\end{aligned} \tag{2.16}$$

$$\begin{aligned}{}_{n+1}H_z(i,j,k)Z_o &= {}_nH_z(i,j,k)Z_o + \frac{1}{\mu_r(i,j,k)}\Big[{}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k) - \\ &\quad {}_{n+\frac{1}{2}}E_x(i,j-\frac{1}{2},k){}_{n+\frac{1}{2}}E_y(i+\frac{1}{2},j,k) + {}_{n+\frac{1}{2}}E_y(i-\frac{1}{2},j,k)\Big]\end{aligned} \tag{2.17}$$

$$\begin{aligned}{}_{n+1}E_x(i,j,k) &= {}_nE_x(i,j,k) + \frac{1}{\epsilon_r(i,j,k)}\Big[{}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o - \\ &\quad {}_{n+\frac{1}{2}}H_z(i,j-\frac{1}{2},k)Z_o - {}_{n+\frac{1}{2}}H_y(i,j,k+\frac{1}{2})Z_o + {}_{n+\frac{1}{2}}H_y(i,j,k-\frac{1}{2})Z_o\Big]\end{aligned} \tag{2.18}$$

$$\begin{aligned}{}_{n+1}E_y(i,j,k) &= {}_nE_y(i,j,k) + \frac{1}{\epsilon_r(i,j,k)}\Big[{}_{n+\frac{1}{2}}H_x(i,j,k+\frac{1}{2})Z_o - \\ &\quad {}_{n+\frac{1}{2}}H_x(i,j,k-\frac{1}{2})Z_o - {}_{n+\frac{1}{2}}H_z(i+\frac{1}{2},j,k)Z_o + {}_{n+\frac{1}{2}}H_z(i-\frac{1}{2},j,k)Z_o\Big]\end{aligned} \tag{2.19}$$

$$\begin{aligned}{}_{n+1}E_z(i,j,k) &= {}_nE_z(i,j,k) + \frac{1}{\epsilon_r(i,j,k)}\Big[{}_{n+\frac{1}{2}}H_y(i+\frac{1}{2},j,k)Z_o - \\ &\quad {}_{n+\frac{1}{2}}H_y(i-\frac{1}{2},j,k)Z_o - {}_{n+\frac{1}{2}}H_x(i,j+\frac{1}{2},k)Z_o + {}_{n+\frac{1}{2}}H_x(i,j-\frac{1}{2},k)Z_o\Big]\end{aligned} \tag{2.20}$$

Equations 2.15 to 2.20 represent the EM field components ($H_x, H_y, H_z, E_x, E_y, E_z$) at a given node (i, j, k) for the next time step in the future (i.e. time=n+1). Equations 2.15 to 2.20 are functions of two things: firstly, the field values at the node now (i.e. time=n) and secondly, the field values $\frac{1}{2}\Delta$ away from the node and half a time step into the future (i.e. time=$n + \frac{1}{2}$). This process is the essence of how the FDTD method calculates the future or predicted field values.

The prediction process can be more clearly understood by referring to Figure 2.1. In the interest of clarity, Figure 2.1 shows only the E field components; the reader must keep in mind the fact that the H field components are also present in the same number and location. The node, (i, j, k), can be thought of as being at the centre of a cube of dimensions $\Delta \times \Delta \times \Delta$. At the start of the process, (time=n), the six field components ($H_x, H_y, H_z, E_x, E_y, E_z$) at the node are known; the three E field components are shown in Figure 2.1(a). The object is to determine what those six values will be at time=$n+1$, the next time step or iteration. What is required is that, at half a time step into the future, the E and H field components located $\frac{1}{2}\Delta$ ahead of, behind, above, below, to the left of, and to the right of the node be determined. These field components can be thought of as being on the surfaces of the cube that encloses the node as shown in Figure 2.1(b). There are 24 such components, twelve E and twelve H. Figure 2.1(b) shows only the twelve E field components.

If the 24 E and H fields components on the cube surfaces can be determined, then all the information required by equations 2.15 to 2.20 will be available and the total $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ fields at the node for the next time step can be calculated (see Figure 2.1(c)).

Let the 24 E and H field components that exist on the surfaces of the nodal cube of Figure 2.1 be collectively known as the "half fields". This is a fitting name because, as will be seen later, they only exist half a time step into the future and half way between adjacent nodes throughout the entire three dimensional problem space. Similarly, let the six E and H field components that exist at the node itself be known as the "node fields".

As mentioned above, to solve equations 2.15 to 2.20, the half fields must be determined. In [17], Chen made use of the unitary condition of the scattering matrix $[S]$ for the TLM symmetrically condensed node (SCN) which is stated as follows:

$$[S][S]^T = 1$$

$[S]^T$ is the transpose of $[S]$. The unitary condition entails that energy within the

time=n

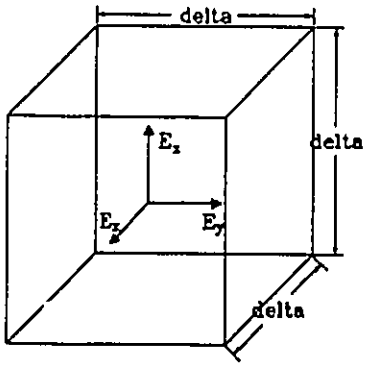


Figure 2.1(a)

time=n+1/2

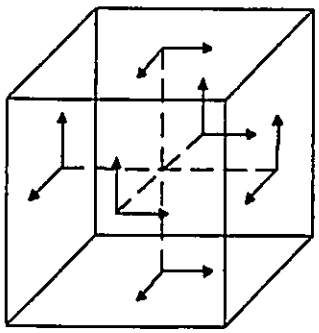


Figure 2.1(b)

time=n+1

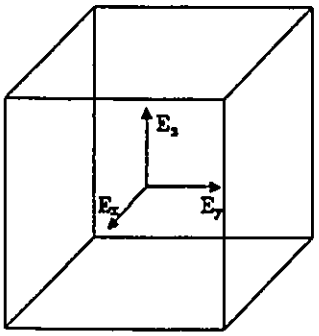


Figure 2.1(c)

Figure 2.1: The FDTD Prediction Process

TLM SCN is conserved; energy is neither created nor destroyed. Energy which enters the TLM SCN at one time step leaves the same TLM SCN at the next time step. In [17], one can multiply out $[S]$ in terms of V^r and V^i , make various substitutions and obtain equations (29) and (30) (which appear below as equations 2.21 and 2.22). Let these equations be known as "boundary equations" because they represent E and H half fields on the surfaces of the nodal cube (see Figure 2.1(b)) which act as a boundary for a single node. The complete set of these boundary equations for the half fields is as follows:

$$ {}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k) - {}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o = \tag{2.21} $$
$$ 2\left[{}_nE_x(i,j,k) - {}_nH_z(i,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_x(i,j-\frac{1}{2},k) - {}_{n-\frac{1}{2}}H_z(i,j-\frac{1}{2},k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o = \tag{2.22} $$
$$ 2\left[{}_nE_x(i,j+1,k) + {}_nH_z(i,j+1,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_x(i,j+\frac{3}{2},k) + {}_{n-\frac{1}{2}}H_z(i,j+\frac{3}{2},k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}H_x(i,j+\frac{1}{2},k)Z_o = \tag{2.23} $$
$$ 2\left[{}_nE_z(i,j,k) + {}_nH_x(i,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_z(i,j-\frac{1}{2},k) + {}_{n-\frac{1}{2}}H_x(i,j-\frac{1}{2},k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) - {}_{n+\frac{1}{2}}H_x(i,j+\frac{1}{2},k)Z_o = \tag{2.24} $$
$$ 2\left[{}_nE_z(i,j+1,k) - {}_nH_x(i,j+1,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_z(i,j+\frac{3}{2},k) - {}_{n-\frac{1}{2}}H_x(i,j+\frac{3}{2},k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_x(i,j,k+\frac{1}{2}) + {}_{n+\frac{1}{2}}H_y(i,j,k+\frac{1}{2})Z_o = \tag{2.25} $$
$$ 2\left[{}_nE_x(i,j,k) + {}_nH_y(i,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_x(i,j,k-\frac{1}{2}) + {}_{n-\frac{1}{2}}H_y(i,j,k+\frac{1}{2})Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_x(i,j,k+\frac{1}{2}) - {}_{n+\frac{1}{2}}H_y(i,j,k+\frac{1}{2})Z_o = \tag{2.26} $$
$$ 2\left[{}_nE_x(i,j,k+1) - {}_nH_y(i,j,k+1)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_x(i,j,k+\frac{3}{2}) - {}_{n-\frac{1}{2}}H_y(i,j,k+\frac{3}{2})Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2}) - {}_{n+\frac{1}{2}}H_x(i,j,k+\frac{1}{2})Z_o = \tag{2.27} $$
$$ 2\left[{}_nE_y(i,j,k) - {}_nH_x(i,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_y(i,j,k-\frac{1}{2}) - {}_{n-\frac{1}{2}}H_x(i,j,k-\frac{1}{2})Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2}) + {}_{n+\frac{1}{2}}H_x(i,j,k+\frac{1}{2})Z_o = \tag{2.28} $$
$$ 2\left[{}_nE_y(i,j,k+1) + {}_nH_x(i,j,k+1)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_y(i,j,k+\frac{3}{2}) + {}_{n-\frac{1}{2}}H_x(i,j,k+\frac{3}{2})Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_y(i+\frac{1}{2},j,k) + {}_{n+\frac{1}{2}}H_z(i+\frac{1}{2},j,k)Z_o = \tag{2.29} $$
$$ 2\left[{}_nE_y(i,j,k) + {}_nH_z(i,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_y(i-\frac{1}{2},j,k) + {}_{n-\frac{1}{2}}H_z(i-\frac{1}{2},j,k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_y(i+\frac{1}{2},j,k) - {}_{n+\frac{1}{2}}H_z(i+\frac{1}{2},j,k)Z_o = \tag{2.30} $$
$$ 2\left[{}_nE_y(i+1,j,k) - {}_nH_z(i+1,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_y(i+\frac{3}{2},j,k) - {}_{n-\frac{1}{2}}H_z(i+\frac{3}{2},j,k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_z(i+\frac{1}{2},j,k) - {}_{n+\frac{1}{2}}H_y(i+\frac{1}{2},j,k)Z_o = \tag{2.31} $$
$$ 2\left[{}_nE_z(i,j,k) - {}_nH_y(i,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_z(i-\frac{1}{2},j,k) - {}_{n-\frac{1}{2}}H_y(i-\frac{1}{2},j,k)Z_o\right] $$

$$ {}_{n+\frac{1}{2}}E_z(i+\frac{1}{2},j,k) + {}_{n+\frac{1}{2}}H_y(i+\frac{1}{2},j,k)Z_o = \tag{2.32} $$
$$ 2\left[{}_nE_z(i+1,j,k) + {}_nH_y(i+1,j,k)Z_o\right] - \left[{}_{n-\frac{1}{2}}E_z(i+\frac{3}{2},j,k) + {}_{n-\frac{1}{2}}H_y(i+\frac{3}{2},j,k)Z_o\right] $$

By manipulating various boundary equations, explicit expressions for the half fields can be obtained. For example, by adding equations 2.21 and 2.22, an explicit expression for ${}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k)$ can be formulated as:

$$ {}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k) = \tag{2.33} $$
$$ {}_nE_x(i,j,k) - {}_nH_z(i,j,k)Z_o + {}_nE_x(i,j+1,k) + {}_nH_z(i,j+1,k)Z_o $$
$$ -\frac{1}{2}\left[{}_{n-\frac{1}{2}}E_x(i,j+\frac{3}{2},k) + {}_{n-\frac{1}{2}}H_z(i,j+\frac{3}{2},k)Z_o \right. $$
$$ \left. + {}_{n-\frac{1}{2}}E_x(i,j-\frac{1}{2},k) - {}_{n-\frac{1}{2}}H_z(i,j-\frac{1}{2},k)Z_o\right] $$

This process is extended to all the boundary equations to form the following set of explicit expressions for the E and H half fields:

$$ {}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o = \tag{2.34} $$
$$ -{}_nE_x(i,j,k) + {}_nH_z(i,j,k)Z_o + {}_nE_x(i,j+1,k) + {}_nH_z(i,j+1,k)Z_o $$
$$ -\frac{1}{2}\left[{}_{n-\frac{1}{2}}E_x(i,j+\frac{3}{2},k) + {}_{n-\frac{1}{2}}H_z(i,j+\frac{3}{2},k)Z_o\right. $$

$$-_{n-\frac{1}{2}}E_x(i,j-\frac{1}{2},k)+_{n-\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o\Big]$$

$$_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k)= \tag{2.35}$$
$$_nE_z(i,j,k)+_nH_x(i,j,k)Z_o+_nE_z(i,j+1,k)-_nH_x(i,j+1,k)Z_o$$
$$-\frac{1}{2}\Big[_{n-\frac{1}{2}}E_z(i,j+\frac{3}{2},k)-_{n-\frac{1}{2}}H_x(i,j+\frac{3}{2},k)Z_o$$
$$+_{n-\frac{1}{2}}E_z(i,j-\frac{1}{2},k)+_{n-\frac{1}{2}}H_x(i,j-\frac{1}{2},k)Z_o\Big]$$

$$_{n+\frac{1}{2}}H_x(i,j+\frac{1}{2},k)Z_o= \tag{2.36}$$
$$_nE_z(i,j,k)+_nH_x(i,j,k)Z_o-_nE_z(i,j+1,k)+_nH_x(i,j+1,k)Z_o$$
$$-\frac{1}{2}\Big[-_{n-\frac{1}{2}}E_z(i,j+\frac{3}{2},k)+_{n-\frac{1}{2}}H_x(i,j+\frac{3}{2},k)Z_o$$
$$+_{n-\frac{1}{2}}E_z(i,j-\frac{1}{2},k)+_{n-\frac{1}{2}}H_x(i,j-\frac{1}{2},k)Z_o\Big]$$

$$_{n+\frac{1}{2}}E_x(i,j,k+\frac{1}{2})= \tag{2.37}$$
$$_nE_x(i,j,k)+_nH_y(i,j,k)Z_o+_nE_x(i,j,k+1)-_nH_y(i,j,k+1)Z_o$$
$$-\frac{1}{2}\Big[_{n-\frac{1}{2}}E_x(i,j,k+\frac{3}{2})-_{n-\frac{1}{2}}H_y(i,j,k+\frac{3}{2})Z_o$$
$$+_{n-\frac{1}{2}}E_x(i,j,k-\frac{1}{2})+_{n-\frac{1}{2}}H_y(i,j,k-\frac{1}{2})Z_o\Big]$$

$$_{n+\frac{1}{2}}H_y(i,j,k+\frac{1}{2})Z_o= \tag{2.38}$$
$$_nE_x(i,j,k)+_nH_y(i,j,k)Z_o-_nE_x(i,j,k+1)+_nH_y(i,j,k+1)Z_o$$
$$-\frac{1}{2}\Big[-_{n-\frac{1}{2}}E_x(i,j,k+\frac{3}{2})+_{n-\frac{1}{2}}H_y(i,j,k+\frac{3}{2})Z_o$$
$$+_{n-\frac{1}{2}}E_x(i,j,k-\frac{1}{2})+_{n-\frac{1}{2}}H_y(i,j,k-\frac{1}{2})Z_o\Big]$$

$$_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2})= \tag{2.39}$$
$$_nE_y(i,j,k)-_nH_x(i,j,k)Z_o+_nE_y(i,j,k+1)+_nH_x(i,j,k+1)Z_o$$
$$-\frac{1}{2}\Big[_{n-\frac{1}{2}}E_y(i,j,k+\frac{3}{2})+_{n-\frac{1}{2}}H_x(i,j,k+\frac{3}{2})Z_o$$
$$+_{n-\frac{1}{2}}E_y(i,j,k-\frac{1}{2})-_{n-\frac{1}{2}}H_x(i,j,k-\frac{1}{2})Z_o\Big]$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}H_x(i,j,k+\frac{1}{2})Z_o = \qquad (2.40)\\
&\quad -{}_nE_y(i,j,k) +{}_nH_x(i,j,k)Z_o +{}_nE_y(i,j,k+1) +{}_nH_x(i,j,k+1)Z_o\\
&\quad -\frac{1}{2}\Big[{}_{n-\frac{1}{2}}E_y(i,j,k+\frac{3}{2}) +{}_{n-\frac{1}{2}}H_x(i,j,k+\frac{3}{2})Z_o\\
&\quad -{}_{n-\frac{1}{2}}E_y(i,j,k-\frac{1}{2}) +{}_{n-\frac{1}{2}}H_x(i,j,k-\frac{1}{2})Z_o\Big]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_y(i+\frac{1}{2},j,k) = \qquad (2.41)\\
&\quad {}_nE_y(i,j,k) +{}_nH_z(i,j,k)Z_o +{}_nE_y(i+1,j,k) -{}_nH_z(i+1,j,k)Z_o\\
&\quad -\frac{1}{2}\Big[{}_{n-\frac{1}{2}}E_y(i+\frac{3}{2},j,k) -{}_{n-\frac{1}{2}}H_z(i+\frac{3}{2},j,k)Z_o\\
&\quad +{}_{n-\frac{1}{2}}E_y(i-\frac{1}{2},j,k) +{}_{n-\frac{1}{2}}H_z(i-\frac{1}{2},j,k)Z_o\Big]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}H_z(i+\frac{1}{2},j,k)Z_o = \qquad (2.42)\\
&\quad {}_nE_y(i,j,k) +{}_nH_z(i,j,k)Z_o -{}_nE_y(i+1,j,k) +{}_nH_z(i+1,j,k)Z_o\\
&\quad -\frac{1}{2}\Big[-{}_{n-\frac{1}{2}}E_y(i+\frac{3}{2},j,k) +{}_{n-\frac{1}{2}}H_z(i+\frac{3}{2},j,k)Z_o\\
&\quad +{}_{n-\frac{1}{2}}E_y(i-\frac{1}{2},j,k) +{}_{n-\frac{1}{2}}H_z(i-\frac{1}{2},j,k)Z_o\Big]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_z(i+\frac{1}{2},j,k) = \qquad (2.43)\\
&\quad {}_nE_z(i,j,k) -{}_nH_y(i,j,k)Z_o +{}_nE_z(i+1,j,k) +{}_nH_y(i+1,j,k)Z_o\\
&\quad -\frac{1}{2}\Big[{}_{n-\frac{1}{2}}E_z(i+\frac{3}{2},j,k) +{}_{n-\frac{1}{2}}H_y(i+\frac{3}{2},j,k)Z_o\\
&\quad +{}_{n-\frac{1}{2}}E_z(i-\frac{1}{2},j,k) -{}_{n-\frac{1}{2}}H_y(i-\frac{1}{2},j,k)Z_o\Big]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}H_y(i+\frac{1}{2},j,k)Z_o = \qquad (2.44)\\
&\quad -{}_nE_z(i,j,k) +{}_nH_y(i,j,k)Z_o +{}_nE_z(i+1,j,k) +{}_nH_y(i+1,j,k)Z_o\\
&\quad -\frac{1}{2}\Big[{}_{n-\frac{1}{2}}E_z(i+\frac{3}{2},j,k) +{}_{n-\frac{1}{2}}H_y(i+\frac{3}{2},j,k)Z_o\\
&\quad -{}_{n-\frac{1}{2}}E_z(i-\frac{1}{2},j,k) +{}_{n-\frac{1}{2}}H_y(i-\frac{1}{2},j,k)Z_o\Big]
\end{aligned}
$$

Consider the twelve half field equations (2.33 to 2.44) that have already been developed. It is certain that they can be solved because in each equation, the only

unknown is the half field expression with the time subscript $n+\frac{1}{2}$. All the other variables in each equation have time subscripts n or $n-\frac{1}{2}$ meaning they are present or past values respectively. In the FDTD simulation process, present field values are known and past field values are stored. Therefore, evaluation of the half field equations and, consequently, equations 2.15 to 2.20 is a straightforward process.

The astute reader will have noted that figure 2.1(b) showed twelve E half fields (and implied twelve more H half fields) located on the surface of the nodal cube while only twelve half field equations were developed (equations 2.33 to 2.44). Further, the observant reader will also have noted that all 24 of these E and H half fields are required to solve equations 2.15 to 2.20. The apparent dilemma is that twelve more half field equations are required for a total of 24 to match the number of half field unknowns in equations 2.15 to 2.20.

To resolve this apparent problem, consider only equation 2.15. Equation 2.15 requires the following half fields:

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2}) \\
&{}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2}) \\
&{}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) \\
&{}_{n+\frac{1}{2}}E_z(i,j-\frac{1}{2},k)
\end{aligned}
$$

Expressions for ${}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2})$ and ${}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k)$ have already been developed (see equations 2.39 and 2.35 respectively). But what of ${}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2})$ and ${}_{n+\frac{1}{2}}E_z(i,j-\frac{1}{2},k)$? No expressions have yet been developed for half fields with negative space arguments (i.e. $(i-\frac{1}{2},j,k)$ or $(i,j-\frac{1}{2},k)$ etc.).

Referring to figure 2.2, assume that (i,j,k) corresponds to the centre of a nodal cube. It can be seen that ${}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2})$ of cube 1 is also ${}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2})$ of cube 2 and because they exist at the same point in space and time, they are equal. Therefore, explicit expressions of half fields with negative space arguments need not be developed; in a sense they already exist. The positive half fields of one cube are also the negative half fields of the cubes adjacent to it. For example, in Figure 2.2, the positive half field for cube 2, ${}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2})$, is also the negative half field for cube 1, ${}_{n+\frac{1}{2}}E_y(i,j,k-\frac{1}{2})$. Similarly, the positive half field for cube 3, ${}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k)$, is also the negative half field for cube 1, ${}_{n+\frac{1}{2}}E_z(i,j-\frac{1}{2},k)$. As a result, all the E half fields for cube 1 are defined and equation 2.15 can be evaluated.

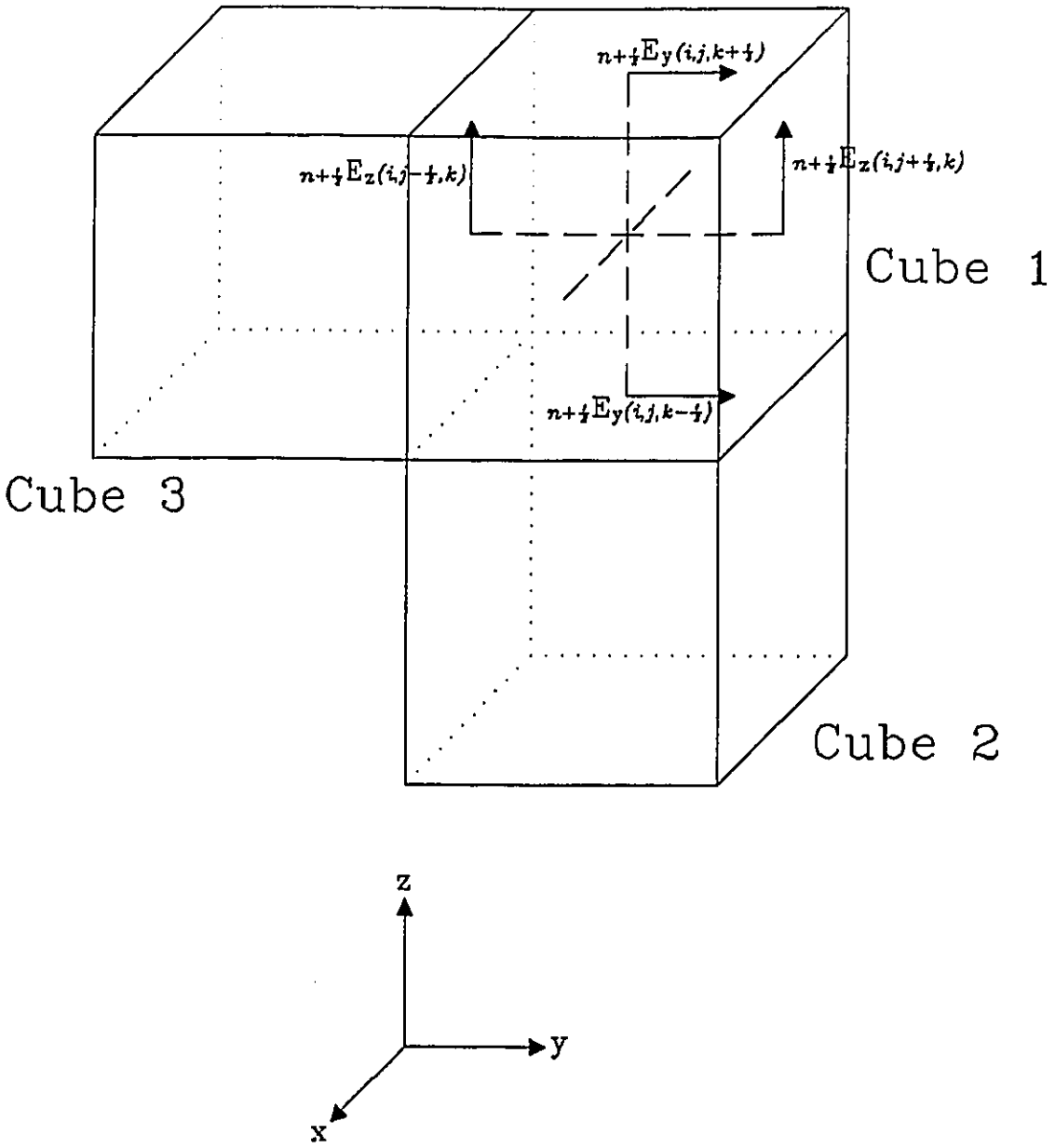


Figure 2.2: Half fields with adjacent nodal cubes.

It is important to bear in mind the nature of the E and H half fields: they are only a step in the FDTD prediction process. The node fields are what is important and it is this information that is sought.

This method does not use a "leap frog" type of algorithm. In the Yee FDTD scheme, E and H field components were calculated on alternating half time steps. At a given time step n, the method determines the half fields at $n + \frac{1}{2}$ throughout the problem space using equations 2.33 to 2.44. Only then can node fields be determined for the next full time step, n+1, using equations 2.15 to 2.20. At this point, the node fields at n and n+1 are known. The process advances to the next time step by replacing half field values at $n - \frac{1}{2}$ with those at $n + \frac{1}{2}$ and node field values at n with those at n+1 throughout the problem space. The process now advances to the next iterative step and half fields at $n + \frac{1}{2}$ and node fields at n+1 are again calculated.

The difference between this method and the FDTD method initially proposed by Yee arises from the way in which the finite difference formulas are applied to equations 2.5 to 2.10. Yee used n and $i + \frac{1}{2}, j + \frac{1}{2}$ or $k + \frac{1}{2}$ (as the case may be) as the centre points whereas this method uses $n + \frac{1}{2}$ and (i, j, k) as its centre points in time and space respectively for the finite differencing operation.

2.3 Boundaries

All simulations of propagating waves and/or scatterers must be bounded in some way due to the finite memories that exist with all computers. The boundaries of the problem space must be carefully chosen so that they simulate, as accurately as possible, the structure under investigation. If the boundaries are incorrectly specified, they may reflect an improper amount of incident energy which would ultimately invalidate the results of any simulation.

In this study, two types of boundaries have been used: the standard electric or E wall boundary and the absorbing wall boundary. For the purposes of this study, all boundaries exist exactly midway between nodes.

The E wall boundary, (henceforth referred to simply as an E wall), is assumed to have infinite conductivity; therefore, all tangential electric field components become zero on its surface. A typical problem space enclosed by an E wall is shown in Figure 2.3. In this case, the E wall encloses the problem space in two dimensions. Assume that the diagram extends to infinity out of the page to realize the third dimension.

At node 1 in Figure 2.3, the E half fields that are tangential to the E wall are set to zero because of the E wall's infinite conductivity. Specifically, the following half fields will be zero:

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_x(i, j+\frac{1}{2}, k) \\
&{}_{n+\frac{1}{2}}E_z(i, j+\frac{1}{2}, k) \\
&{}_{n+\frac{1}{2}}E_x(i, j, k+\frac{1}{2}) \\
&{}_{n+\frac{1}{2}}E_z(i, j, k+\frac{1}{2})
\end{aligned}
$$

(i, j, k) refers to node 1 as indicted in Figure 2.3. If (i, j, k) referred to node 2, then only the first two field components mentioned above would be zero.

Of course, no such simplifying assumption can be made about the H half fields that are tangential to the E wall. In fact, using the half field equations (2.33 to 2.44), it would appear that a problem exists when evaluating H half field expressions. As an example of this, consider Figure 2.3 with (i, j, k) referenced to node 1. Equation 2.34 shows how ${}_{n+\frac{1}{2}}H_z(i, j+\frac{1}{2}, k)Z_o$ is to be calculated. The problem arises when terms with space arguments that exist outside the E wall are required. In this example, equation 2.34 requires the following node and half fields that exist beyond the E wall:

$$
\begin{aligned}
&{}_nE_x(i, j+1, k) \\
&{}_nH_z(i, j+1, k)Z_o \\
&{}_{n-\frac{1}{2}}E_x(i, j+\frac{3}{2}, k) \\
&{}_{n-\frac{1}{2}}H_z(i, j+\frac{3}{2}, k)Z_o
\end{aligned}
$$

A boundary marks the limit of the problem space. While half fields can exist on its surface, nothing exists beyond it. The question then is how to solve for H half fields that are tangential to E walls. As shown above, the regular half field equations cannot be applied because they reference nodes that exist beyond the E wall. All is not lost; the reader may recall the tedious boundary equations (equations 2.21 to 2.32) of chapter 2.2. When (i, j, k) refers to nodes adjacent to E walls, these equations can be used directly to determine the tangential H half fields. In this case, ${}_{n+\frac{1}{2}}H_z(i, j+\frac{1}{2}, k)Z_o$ is to be calculated by using equation 2.21 knowing that ${}_{n+\frac{1}{2}}E_x(i, j+\frac{1}{2}, k)$ is zero. The expression becomes:

$${}_{n+\frac{1}{2}}H_z(i, j+\frac{1}{2}, k)Z_o =$$

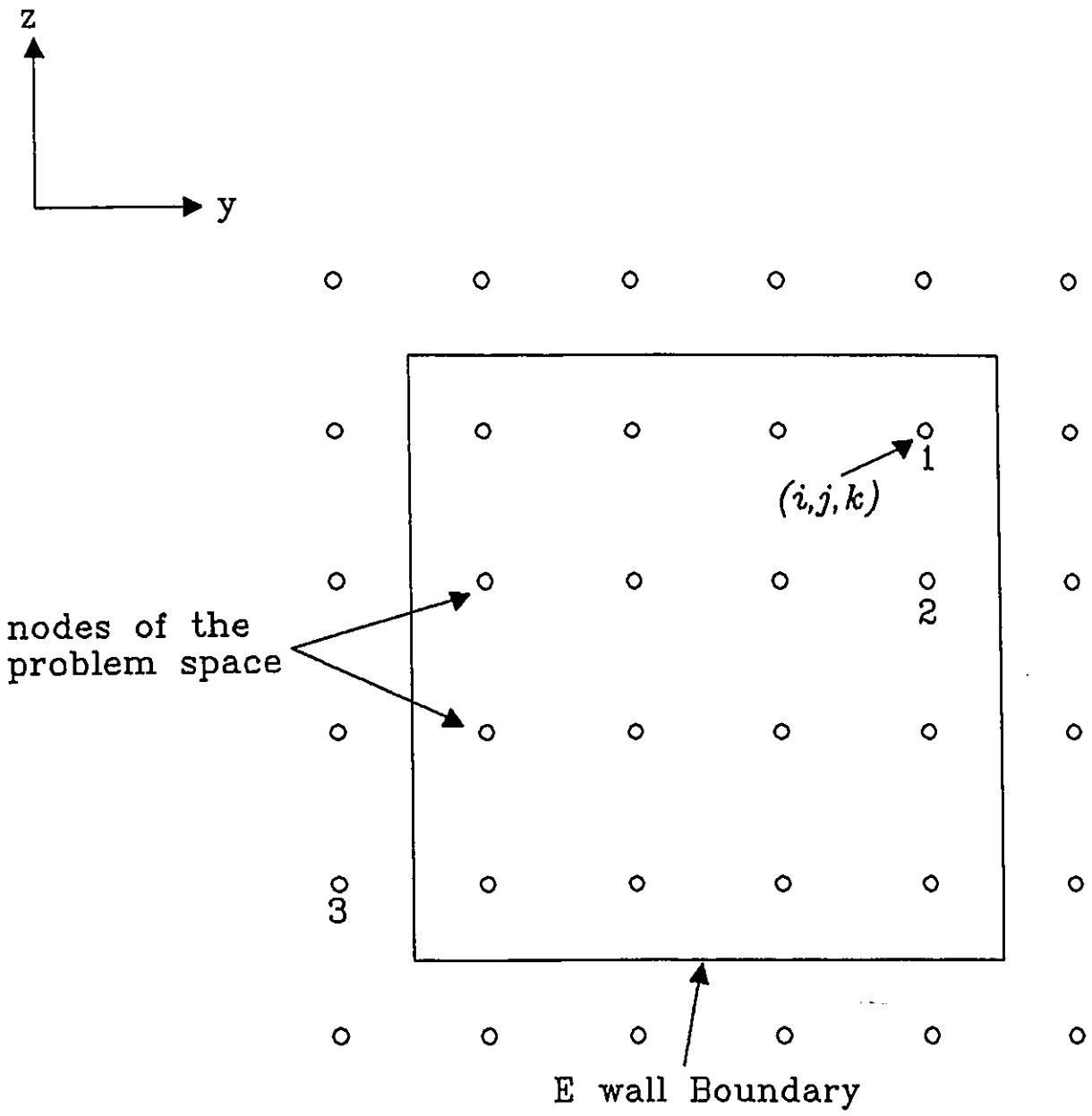


Figure 2.3: Treatment of half fields tangential to E walls

$$-2\left[{}_{n}E_{x}(i,j,k)-{}_{n}H_{z}(i,j,k)Z_{o}\right]+\left[{}_{n-\frac{1}{2}}E_{x}(i,j-\frac{1}{2},k)-{}_{n-\frac{1}{2}}H_{z}(i,j-\frac{1}{2},k)Z_{o}\right]$$

The above expression can be readily evaluated because all the node and half fields required exist within the problem space as defined in Figure 2.3.

Suppose (i,j,k) referred to node 3. Although this node is not in the problem space, E and H half fields along the y axis (which will be tangential to the E wall) will be. As before, tangential E half fields will be zero. To determine the H half fields, the boundary equations must again be used. For example, assume that ${}_{n+\frac{1}{2}}H_{z}(i,j+\frac{1}{2},k)Z_{o}$ is to be determined. In this case, equation 2.21 cannot be used because it would require node and half field components that would be outside the problem space. Equation 2.22 ought to be used instead. Setting ${}_{n+\frac{1}{2}}E_{x}(i,j+\frac{1}{2},k)$ to zero, equation 2.22 becomes:

$${}_{n+\frac{1}{2}}H_{z}(i,j+\frac{1}{2},k)Z_{o}=$$
$$2\left[{}_{n}E_{x}(i,j+1,k)+{}_{n}H_{z}(i,j+1,k)Z_{o}\right]-\left[{}_{n-\frac{1}{2}}E_{x}(i,j+\frac{3}{2},k)+{}_{n-\frac{1}{2}}H_{z}(i,j+\frac{3}{2},k)Z_{o}\right]$$

Once again, the above expression can be evaluated because it references only node and half fields that exist within the problem space.

Magnetic or M wall boundaries and their accompanying equations could also be developed in a similar manner with the difference being that tangential H half fields would be set to zero. This study does not make use of this type of wall so its development will not be pursued.

In summary, when dealing with nodes that are adjacent to E walls, there are only two things to bear in mind: firstly, all E half fields that are tangential to the E wall are set to zero and, secondly, H half fields are calculated by selecting the appropriate boundary equation (equations 2.21 to 2.32). The choice of which boundary equation to use (there will only be two to choose from) arises from the location of the node with respect to the E wall; the node being either just outside or inside the problem space.

The second type of boundary used in this study is known as an absorbing wall (AW) boundary. An AW is meant to absorb all incident energy without generating any reflections. The AWs are used to limit the problem space when simulating radiation or propagation in a free space environment.

The AW used in this study is based upon Chen's modification [21] to Saguet's absorbing boundary for three nodes [22]. Chen's AW boundary represents an improve-

ment upon that of Saguet in that it provides some suppression of spurious modes as well as absorbing the physical modes.

Spurious modes exist only in numerical simualtions involving a discretized node mesh or grid, they do not occur in the physical structure. In [23], Nielsen descibes these modes as occuring when the structure being simulated has features in the order of several lattice (Δ in this case) spacings.

To better understand the workings of the AW, consider Figure 2.4 which shows a single row of nodes, along the y axis, from a 3D problem space. The row ends in an AW. In order to predict the E_x and H_z half fields that would exist at $n+\frac{1}{2}$ and $j+\frac{1}{2}$ (i.e. on the surface of the AW) the following AW "boundary equation" must be used:

$$
\begin{aligned}
{}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o = \\
2.25\left[{}_{n-\frac{7}{2}}E_x(i,j-\frac{3}{2},k) + {}_{n-\frac{7}{2}}H_z(i,j-\frac{3}{2},k)Z_o\right] \\
-1.5\left[{}_{n-\frac{15}{2}}E_x(i,j-\frac{7}{2},k) + {}_{n-\frac{15}{2}}H_z(i,j-\frac{7}{2},k)Z_o\right] \\
0.25\left[{}_{n-\frac{23}{2}}E_x(i,j-\frac{11}{2},k) + {}_{n-\frac{23}{2}}H_z(i,j-\frac{11}{2},k)Z_o\right]
\end{aligned}
\tag{2.45}
$$

By adding equations 2.45 and 2.21 (the appropriate boundary equation), an explicit expression for ${}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k)$, an AW half field equation, can be obtained. Similarly, by subtracting equation 2.45 from 2.21, one obtains an explicit expression for ${}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o$. The complete set of AW boundary equations can be found in appendix A. The regular half field equations (2.33 to 2.44) could not have been used to generate explicit expressions for half fields on the surface of AWs because, as was the case with the E wall, they referenced nodes beyond the AW. The boundary equations (2.21 to 2.32) alone could not have been resorted to because neither of the half fields tangential to the AW could have been set to zero (as was the case with the E wall). Therefore, the development of the AW boundary equations was necessary. Of course, this approach can be extended to AWs which are perpendicular to x and z axes as well.

In Figure 2.4, it is assumed that at time n, the values for the E_x and H_z half fields located at $j+\frac{1}{2}$ from the node indicated by (i,j,k) (i.e. on the surface of the AW), are to be predicted by equations 2.45 and 2.21. In order to do this, the values for the half fields from previous nodes and times steps must be known. In this specific example, half fields E_x and H_z at locations $j-\frac{3}{2}$, $j-\frac{7}{2}$, and $j-\frac{11}{2}$ at previous time steps $n-\frac{7}{2}$, $n-\frac{15}{2}$ and $n-\frac{23}{2}$ respectively, must all be known. Thus, the predicted

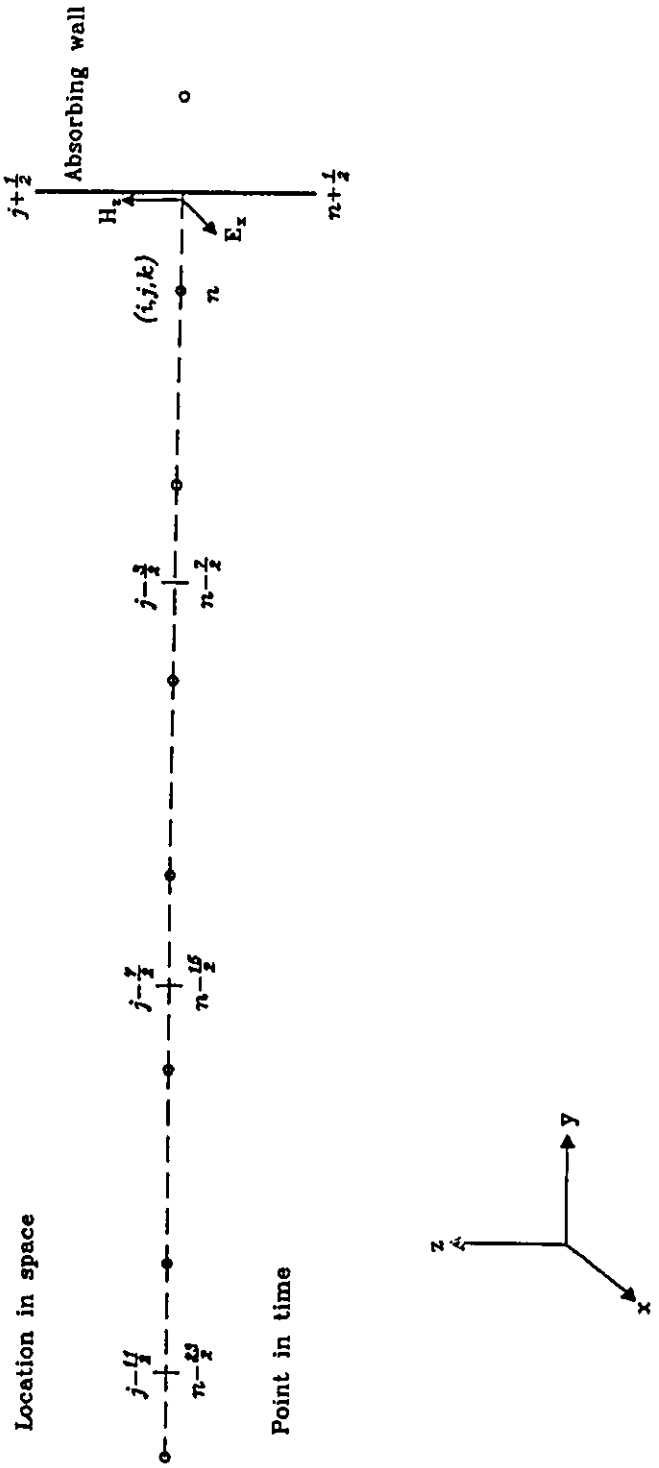


Figure 2.4: Half field calculation for the absorbing wall.

values for the half fields on the surface of an AW depend upon past half field values at locations up to six nodes away.

2.4 Stability

The node field equations (2.15 to 2.20) combined with the half field equations (2.33 to 2.44) represent a time-stepping algorithm of EM energy propagation through a medium characterized by μ and ϵ. The numerical stability of the algorithm will depend upon the values assigned to $\Delta x, \Delta y, \Delta z$ and Δt.

This study makes use of the stability criterion developed by Taflove and Brodwin [19] as follows:

$$v_{max}\Delta t \leq \left[\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right]^{-\frac{1}{2}} \tag{2.46}$$

v_{max} is the maximum wave phase velocity expected in the model.

$$v_{max} = \frac{c_o}{\sqrt{\mu_r \epsilon_r}}$$

For the ease of calculation, the grid spacing parameters are chosen such that $\Delta = \Delta x = \Delta y = \Delta z$. Therefore, equation 2.46 can be rewritten as:

$$\begin{aligned} \frac{c_o}{\sqrt{\mu_r \epsilon_r}}\Delta t &\leq \left[\frac{3}{\Delta^2}\right]^{-\frac{1}{2}} \\ \frac{c_o}{\sqrt{\mu_r \epsilon_r}}\frac{\Delta t}{\Delta} &\leq \frac{1}{\sqrt{3}} \end{aligned} \tag{2.47}$$

For simplicity, Δt and Δ have been selected such that $\frac{\Delta}{\Delta t} = c_o$. Consequently, "free space" shall be characterized by $\mu_r = \epsilon_r = 2$ in order that the inequality of 2.47 be respected.

The stability criterion, equation 2.46, was originally developed for use with the Yee FDTD algorithm. In the appendix of [19], the authors provide the derivation of 2.46 which is independent of the manner in which the finite difference equations are applied. Therefore, the stability criterion developed by Taflove and Brodwin is as applicable to this method as it was to Yee's.

Therefore, equation 2.47 can be rewritten as:

$$0.5 \leq 0.577 \tag{2.48}$$

The choices made for $\mu_r, \epsilon_r, \Delta$ and Δt, satisfy the stability criterion as expressed by the inequality 2.48. Nielson and Hoefer [28] showed that the numerical dispersion for a symmetrically condensed node TLM mesh did not vary significantly from the ideal case even for $k_o\Delta$ as high as 0.9. Since this method is equivalent to the symmetrical condensed TLM node (as established by Chen *et al.* [17]), it can be concluded that, given the choices made for Δ and frequency (8-12 GHz), this method should not suffer significantly from numerical dispersion. The worst case being,

$$k_o\Delta = 2\pi\frac{\Delta}{\lambda} = \frac{2\pi\Delta}{0.0375} = 0.1676$$

Furthermore, the choices made are in accordance with Kim's and Zhang and Mei's recommendations ([4],[15]) that one should operate near the upper limit of the stability inequality (2.48) to minimize the numerical dispersion and obtain high accuracy.

2.5 FDTD Example: Resonant Cavity

In order to provide the reader with some confidence in the method, a simulation of a resonant cavity was performed. The cavity dimensions are as shown in Figure 2.5. The cavity is constructed of E walls which enclose a medium characterized by μ_r=1 and ϵ_r=2 (in the actual simulation, μ_r=2 and ϵ_r=4 so that the inequality 2.48 be satisfied). The excitation is a spatially-distributed half sinusoid (as used for TE_{10} waveguide mode), time-dependent Gaussian pulse. This type of excitation is discussed in greater detail in chapter 3.2.

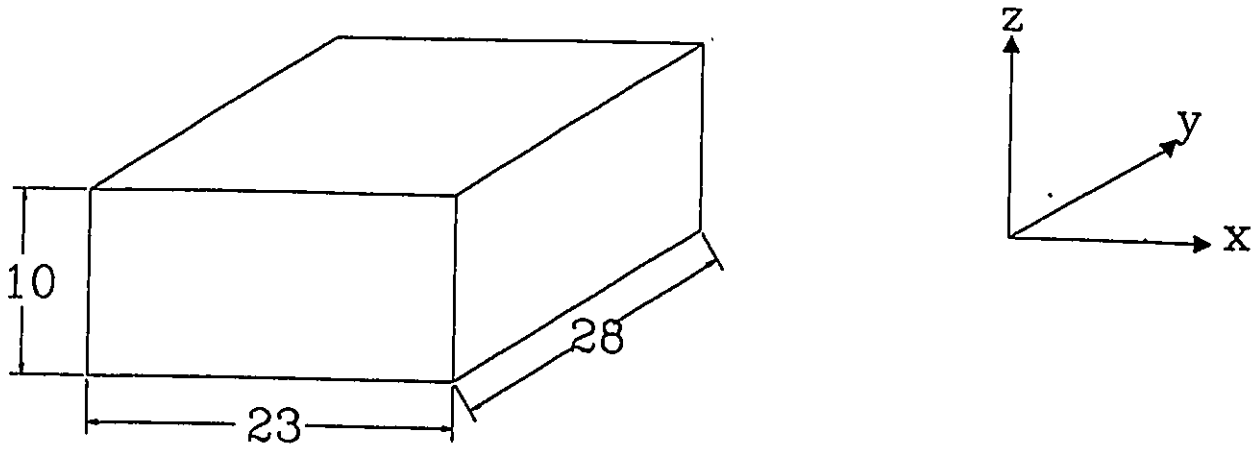
The resonant frequency for the TE_{101} mode (the lowest order mode for this particular structure) is given by:

$$\begin{aligned} f_{res}[TE_{101}] &= \frac{1}{2\sqrt{\mu\epsilon}}\sqrt{\frac{1}{a^2}+\frac{1}{d^2}} \\ &= \frac{c_o}{2\sqrt{\mu_r\epsilon_r}}\sqrt{\frac{1}{a^2}+\frac{1}{d^2}} \\ &= 5.968\,GHz \end{aligned}$$

The simulation was run for 1000 iterations. The E_z field component was extracted from the node $(i,j,k) = (12,15,5)$ in a $23\Delta\times28\Delta\times10\Delta$ problem space and Fourier transformed by an FFT algorithm (details of the algorithm will be discussed later). The magnitude spectrum of the E_z data from the simulation is shown in Figure 2.6.

The resonant peak for the TE_{101} mode is clearly visible as predicted by the above calculations. Also apparent is the peak for the TE_{103} mode, f_{res}=12.264 GHz. Other modes (i.e. TE_{110}, TE_{011}... etc.) are also present in Figure 2.6 but, due to the location where E_z data is extracted, their peaks are not as pronounced.

Thus it may be concluded that, given the location from which data has been extracted, the method has accurately predicted cavity resonant frequencies. This brief example should provide the skeptical reader with some measure of confidence that, thus far, the simulation is valid and the method reliable.



All dimensions in mm
a=23mm
d=28mm

Figure 2.5: FDTD simulation of a resonant cavity

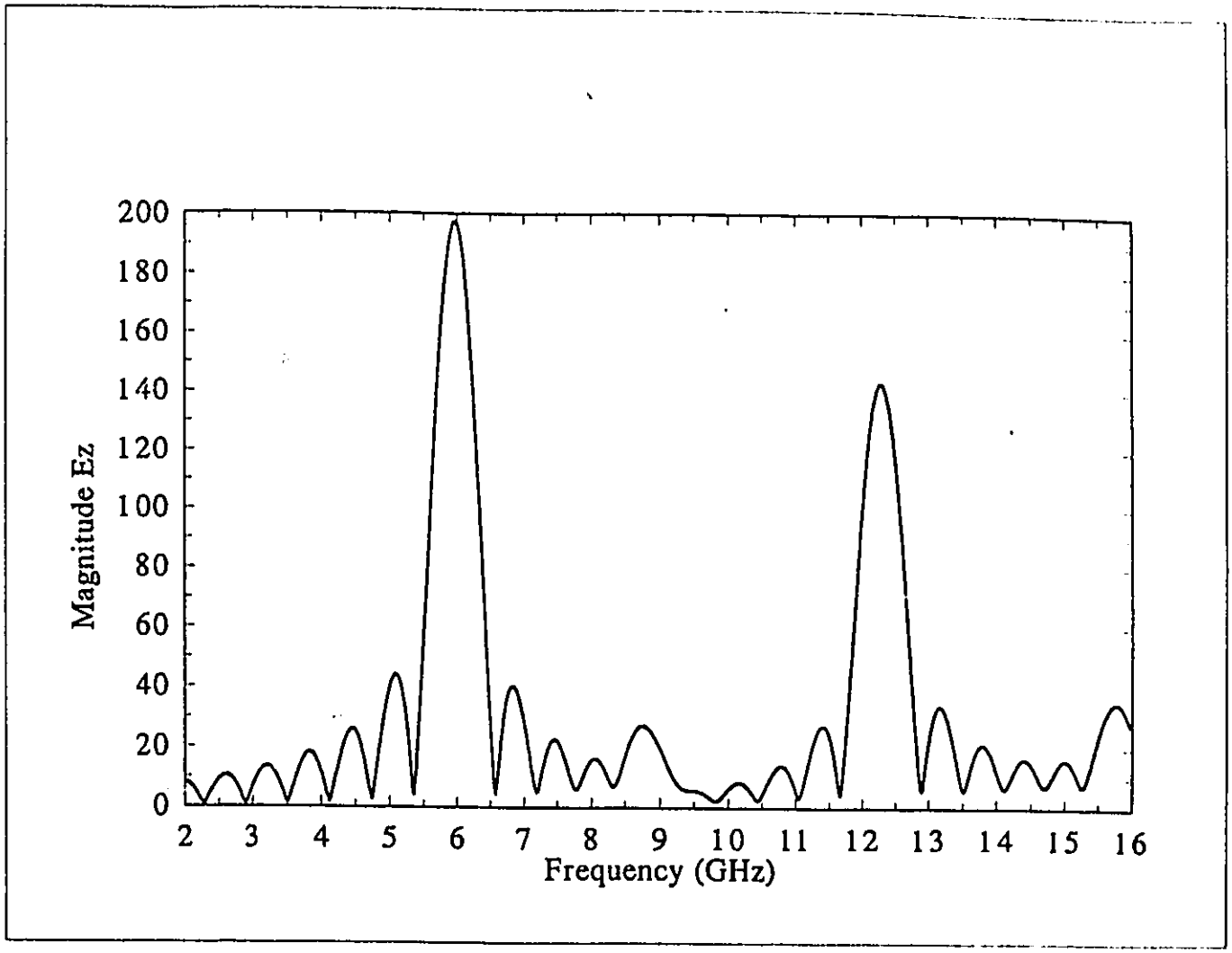


Figure 2.6: Magnitude spectrum of E_z data

Chapter 3

The FDTD Simulation

3.1 Introduction

This chapter will discuss the specifics of the FDTD simulation as it applies to this study. Figure 3.1 provides an exploded view of the physical structure to be simulated: the EXPAR device. EXPAR consists of three components: a length of X-band waveguide (WR90), a resonant iris, a dielectric covering.

The WR90 waveguide is 50 mm long and has internal dimensions of 10.16×22.86 mm and a wall thickness of 1.27 mm [24, page 289]. The resonant iris is 0.2 mm thick and extends 2 mm into the waveguide from each of the interior walls. The iris is mounted in such a manner as to provide continuous electrical contact with the outer walls of the waveguide. These details can be clearly seen by referring to the cross-sectional diagrams of the EXPAR device in Figure 3.2.

The dielectric is RT/duriod 5870 with $\mu_r = 1$ and $\epsilon_r = 2.33 \pm .02$ @ 10GHz. Its dimensions are 10×23 mm with a thickness of 1.6 mm. As can be seen from Figure 3.2, it does not overlap significantly upon the walls of the waveguide.

For the purposes of simulating the EXPAR device, the following three structures were used: EXPAR1 which consisted only of the 50 mm length of waveguide, EXPAR2 which was composed of the waveguide section plus the iris, EXPAR3 which was the complete structure (waveguide, iris and dielectric).

In the simulation of each structure, one end of the device (the reference plane) is connected to a 70 mm length of identical waveguide which terminates at an AW. This section contains the plane of excitation and the region over which the standing-wave pattern will be determined. The other end of each structure (the aperture) radiates into a free space region bounded by AWs. The walls of the waveguide and iris are

composed of E walls.

In the simulation, the following approximations have been made: The waveguide is assumed to have internal dimensions of 10×23 mm and a wall thickness of 1 mm. The iris has negligible thickness and the dielectric is 2 mm thick. Figure 3.3 shows a cross-sectional top view of the three structures and their simulation configurations. All calculations for the reflection coefficient (Γ) are with respect to the reference plane, looking into the device. Note that the flange of the original EXPAR device has not been modelled; the impact of this will be discussed later on. The dimensions of the free space cavity are 43 × 43 × 43 mm. The author assumes that the reader understands that Γ is a function of frequency and therefore, in the interest of brevity and clarity, he declines to write its arguement at each instance except when stressing this dependence.

The determination of Γ for the EXPAR device was divided into three stages. Each stage involved the comparison of simulated and measured data of Γ for each of the three structures: EXAPR1, EXPAR2 and EXPAR3. All measurements were performed by using an HP8510B Network Analyzer. It was the author's experience that, once the Network Analyzer was properly calibrated, measurement of Γ was a straightforward process. The same cannot be said about the simulation; the numerical analysis required to determine Γ was more involved and will be discussed in chapter 3.3.

In each stage of the simulation, an excitation was applied (which started the whole process) and allowed to run for 1000 iterations. During the simulation, data was collected over a specified region and saved for later Fourier analysis. Once data had been gathered and transformed to the frequency domain, Γ could then be calculated. The approach taken for exciting the simulation, performing the Fourier analysis and calculating Γ will be discussed first before presenting any data from either the simulation or the measurements. A discussion of the discrepancies between measured and simulated data will conclude the chapter.

3.2 Excitation

As already mentioned in chapter 2.5, the excitation used with the three EXPAR structures is a spatially-distributed half sinusoid, time-dependent Gaussian pulse. This type of distribution excites the TE_{10} mode (i.e. the dominant mode) in the

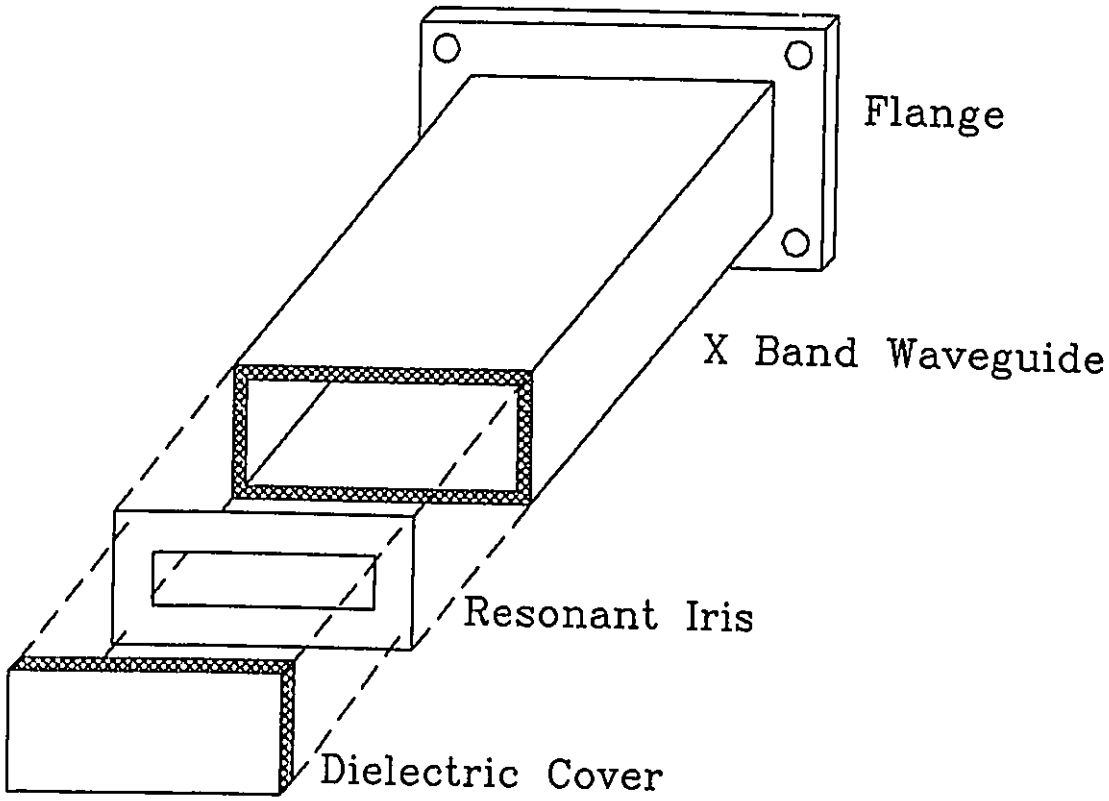


Figure 3.1: Exploded view of EXPAR device

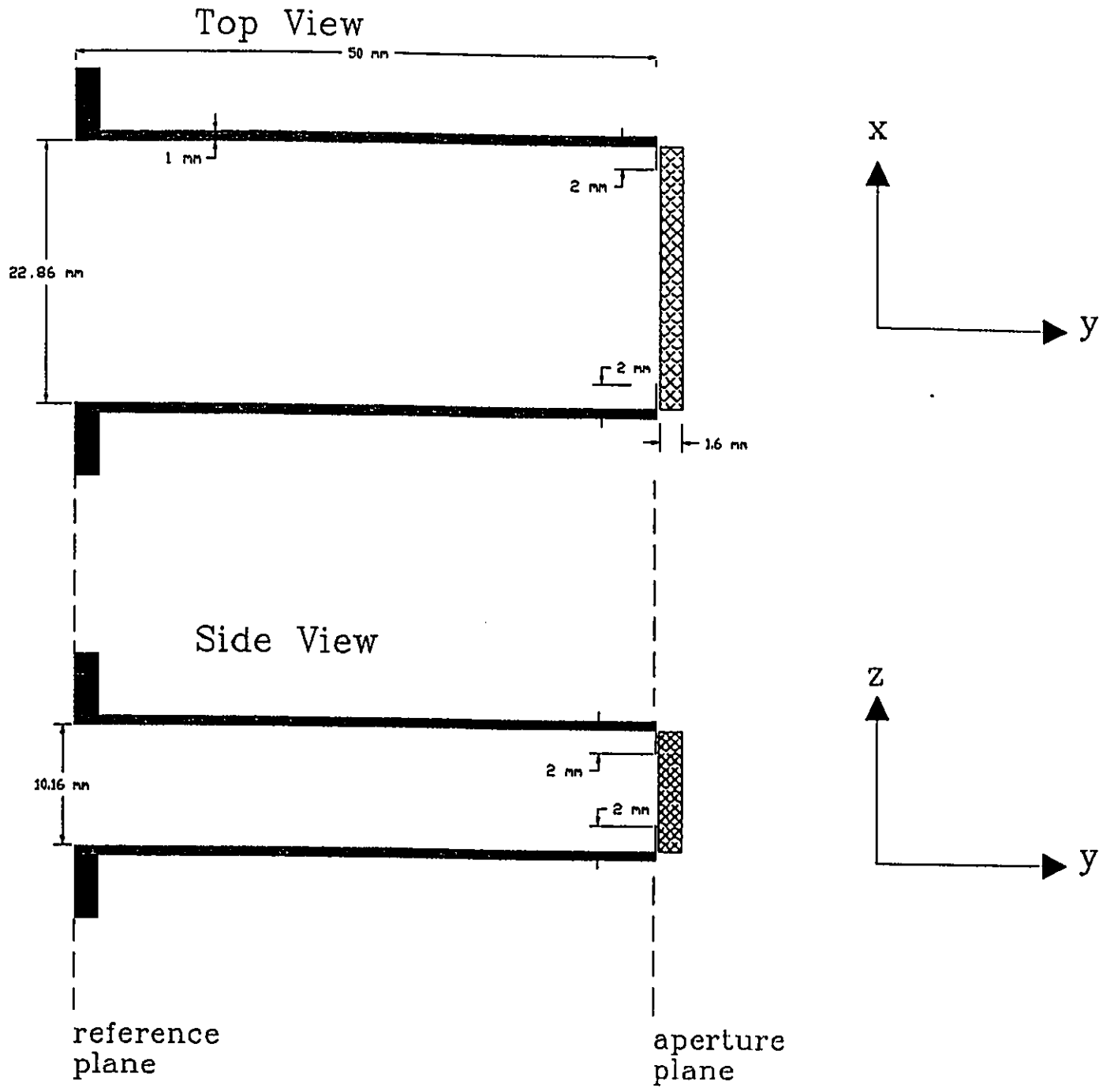


Figure 3.2: Cross-sectional views of EXPAR device

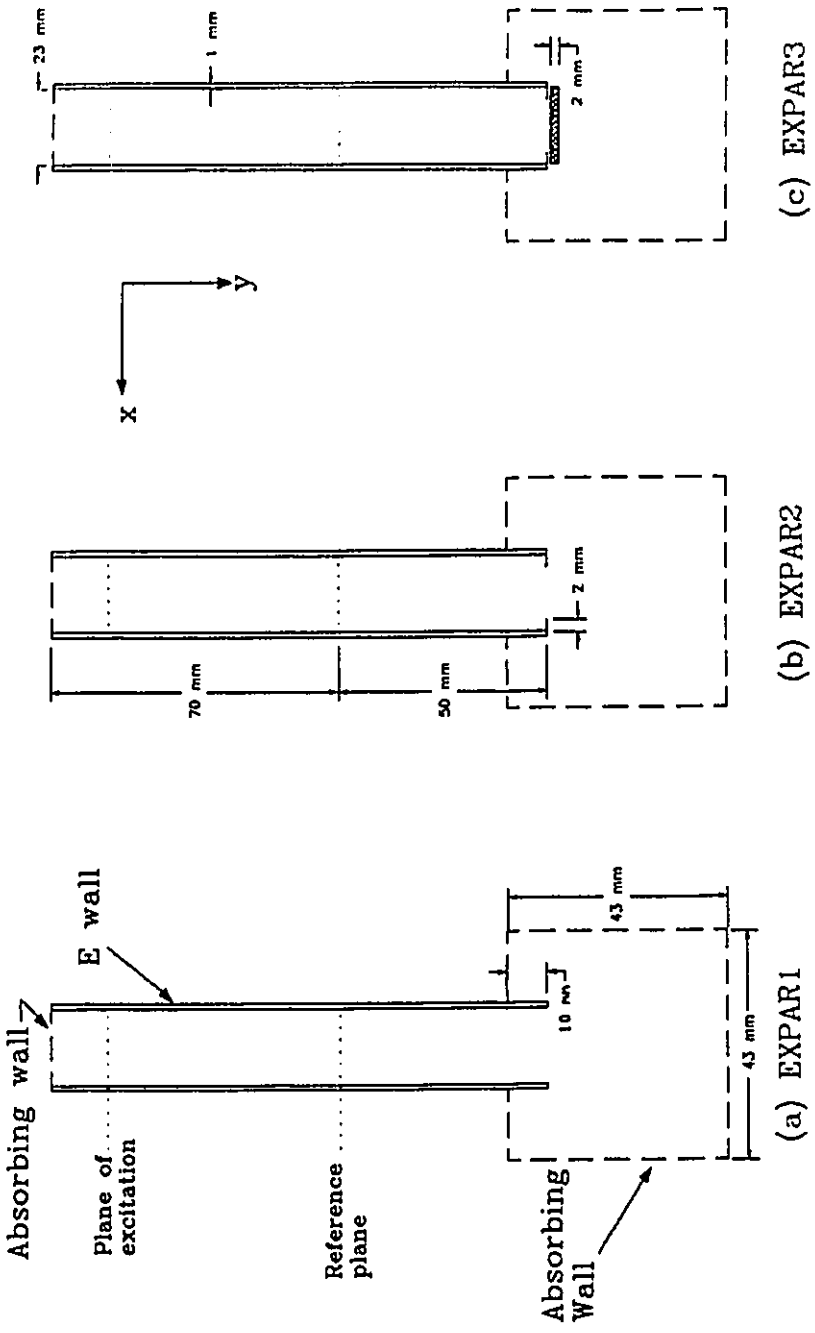


Figure 3.3: Cross-sectional top view of EXPAR1, EXPAR2 and EXPAR3

guide.

The excitation plane is at j=15 where j=1 represents the first plane of nodes in from the AW as can be seen from Figure 3.4 Note that j=0 represents the plane of nodes that exist on the other side of the AW which is located midway between the j=0 and j=1 planes. Figure 3.4 represents the excitation of any of the three EXPAR structures of Figure 3.3. The excitation is accomplished by injecting E_z and H_x half fields into the calculation process midway between the plane of nodes at j=15 and j=14.

Half field injection occurs only when the node j index is equal to 15 within the waveguide. When the computer calculates the values for E_z and H_x node fields at nodes on the excitation plane for the next time step, it requires (among other things) the values for the E_z and H_x half fields to the left and the right of the node (i.e. on either side of the node). The excitation is added to the half field that would already exist on the left side of the node. This method of addition or injection allows an excitation to be introduced into the problem space without affecting the natural flow or scattering of energy throughout the mesh as was done in [19].

By adding the excitation only to the half fields on the left of the nodes of the excitation plane, most of the energy (but not all, some goes to the left) is observed to propagate to the right. If the excitation were added only to the half fields on the right, most of the energy would propagate to the left. If the excitation were added equally to both, the energy propagation would be symmetrical: an equal amount propagating to the left and to the right.

The amount of excitation that is added depends upon time (i.e. the iteration or step number in the program) and location within the excitation plane. The equations that regulate the excitation are as follows:

$$ {}_{n+\frac{1}{2}}E_z^{ex}(i,15,k) = \sin\left[\frac{\pi(i-\frac{1}{2})}{23}\right] e^{-\left(\frac{n-40}{10}\right)^2} \tag{3.1} $$

$$ {}_{n+\frac{1}{2}}H_x^{ex}(i,15,k) = \sin\left[\frac{\pi(i-\frac{1}{2})}{23}\right] e^{-\left(\frac{n-40}{10}\right)^2} \tag{3.2} $$

Where n=1,2,3...1000. Equations 3.1 and 3.2 introduce E_z^{ex} and H_x^{ex} half fields in a half sinusoid along the width of the guide. The magnitude of the excitation grows with time; at the fortieth iteration it is at its peak and it declines thereafter. The reader will recall that each iteration or time step (n) in the simulation represents 3.33333 psec. of elapsed real time. The time variation of ${}_{n+\frac{1}{2}}E_z^{ex}(i,15,k)$ (equation

3.1) is shown in Figure 3.5 (it is identical to that of ${}_{n+\frac{1}{2}}H_x^{ex}(i,15,k)$). The spectrum of equation 3.1 is $E_z^{ex}(f)$ and is given by:

$$E_z^{ex}(f) = \frac{10}{\sqrt{2}}\, e^{j0.838f}\, e^{-\frac{1}{2}\left(\frac{f}{6.75}\right)^2} \tag{3.3}$$

In equation 3.3, f is in GHz. A graphical representation of the normalized magnitude of equation 3.3 is provided in Figure 3.6.

In equation 3.3, the standard deviation (σ) is 6.75. It is generally accepted that, for Gaussian distributions, 68% of the values lie below σ and 32% lie above it. The TE_{10} cut-off frequency for WR90 is 6.562 GHz. Therefore, in this case, a significant amount of energy (about 32% of the frequency content of equation 3.3) will propagate in the WR90 waveguide. Some of the energy which exists below the cut-off frequency propagates a short distance in evanescent TE_{10} modes. The rest is stored in the region of the plane of excitation and, as was observed in the simulation, oscillates with a high rate of decay.

3.3 Method for Calculating Γ

The calculation of Γ for the three EXPAR structures has little to do with the FDTD simulation *per se*. It does, however, demonstrate one of the many ways that EM field simulation data can be used to extract meaningful information about a specific microwave structure.

The method for calculating Γ is based upon determining the standing-wave pattern over a region in the waveguide which extends from j=20 to the reference plane (j=70) as shown in Figure 3.8. Γ is determined looking into the reference plane along the y axis in the direction of increasing j.

Consider a plane of nodes at j=20 which forms the cross-section of the problem space in the waveguide region as shown in Figure 3.7. The E_z node field (i.e. the E_z field component at the node) will be used in this procedure since it is the most prevalent of the six possible node fields due to the manner in which the structure was excited. Furthermore, only the dominant mode (TE_{10}) will be considered. In [27, page 458], the Fourier transform of a time signal, $f(t)$, is given by:

$$f(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

Let $A^{TE10}(j,t)$ be the time-dependent weighting coefficient for the TE_{10} mode. That is to say $A^{TE10}(j,t)$ dictates "how much" E_z node field exists exclusively in the TE_{10}

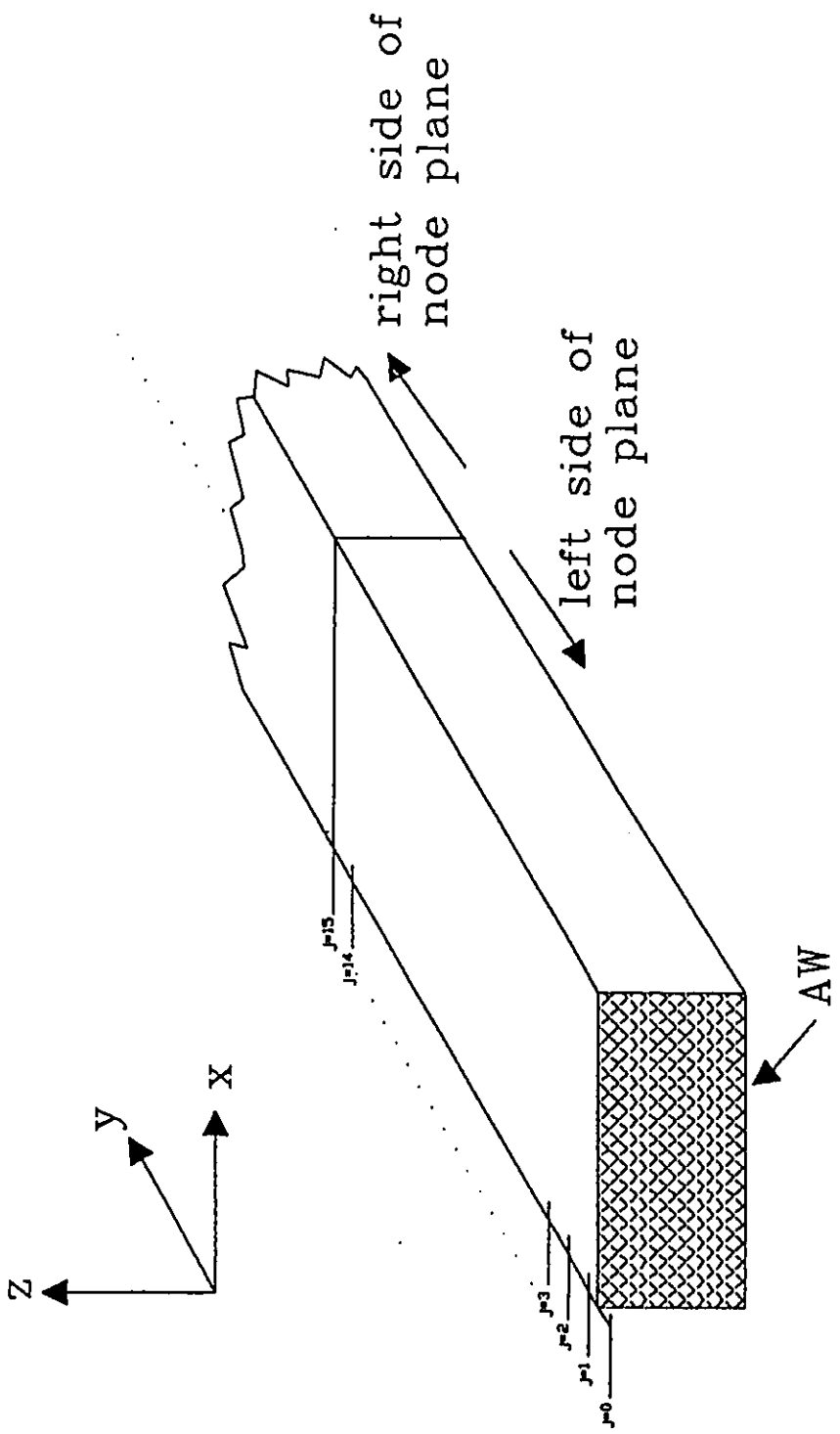


Figure 3.4: Excitation plane of waveguide section

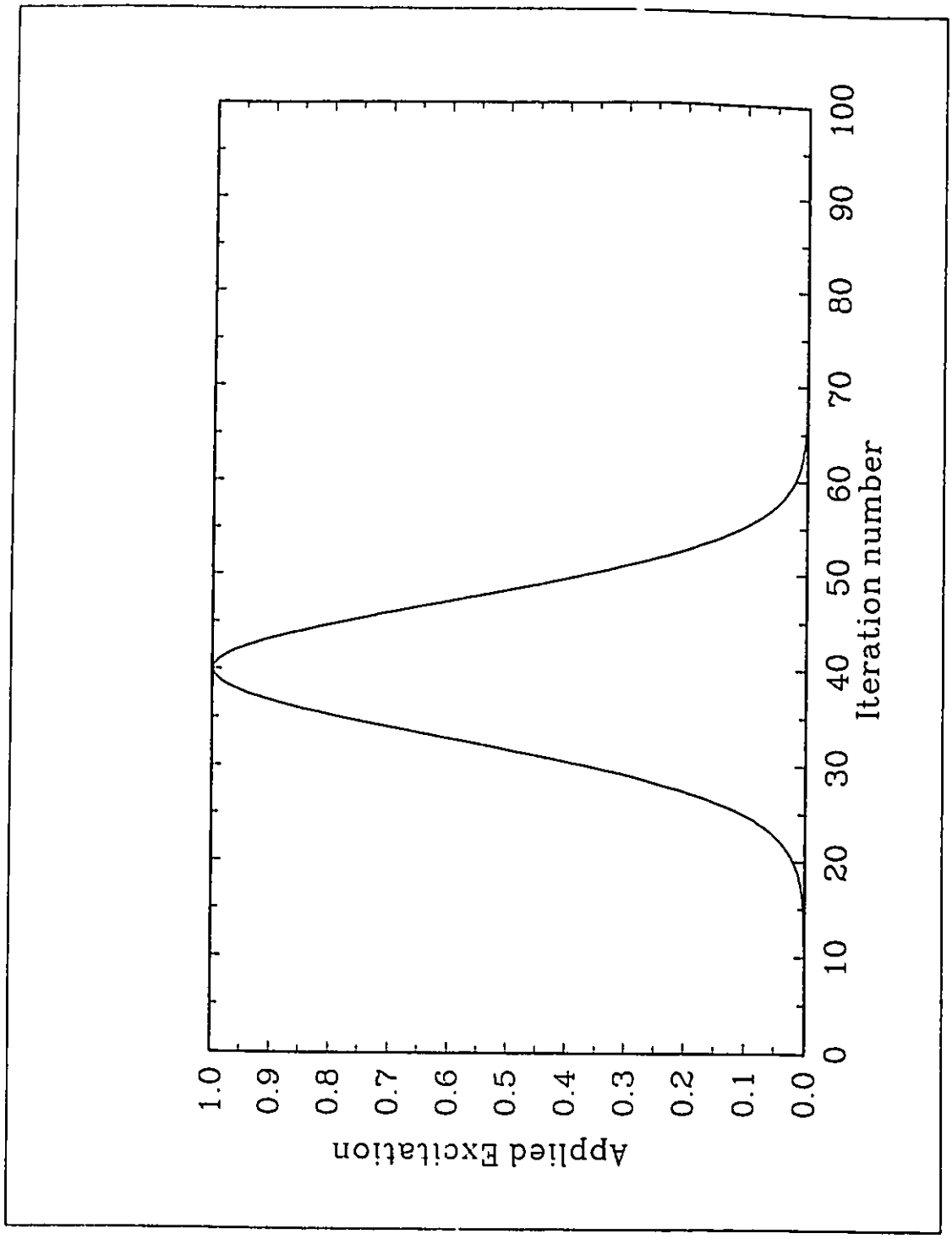


Figure 3.5: Excitation injected into simulation, ${}_{n+\frac{1}{2}}E_z^{ex}(i,15,k)$ or ${}_{n+\frac{1}{2}}H_x^{ex}(i,15,k)$

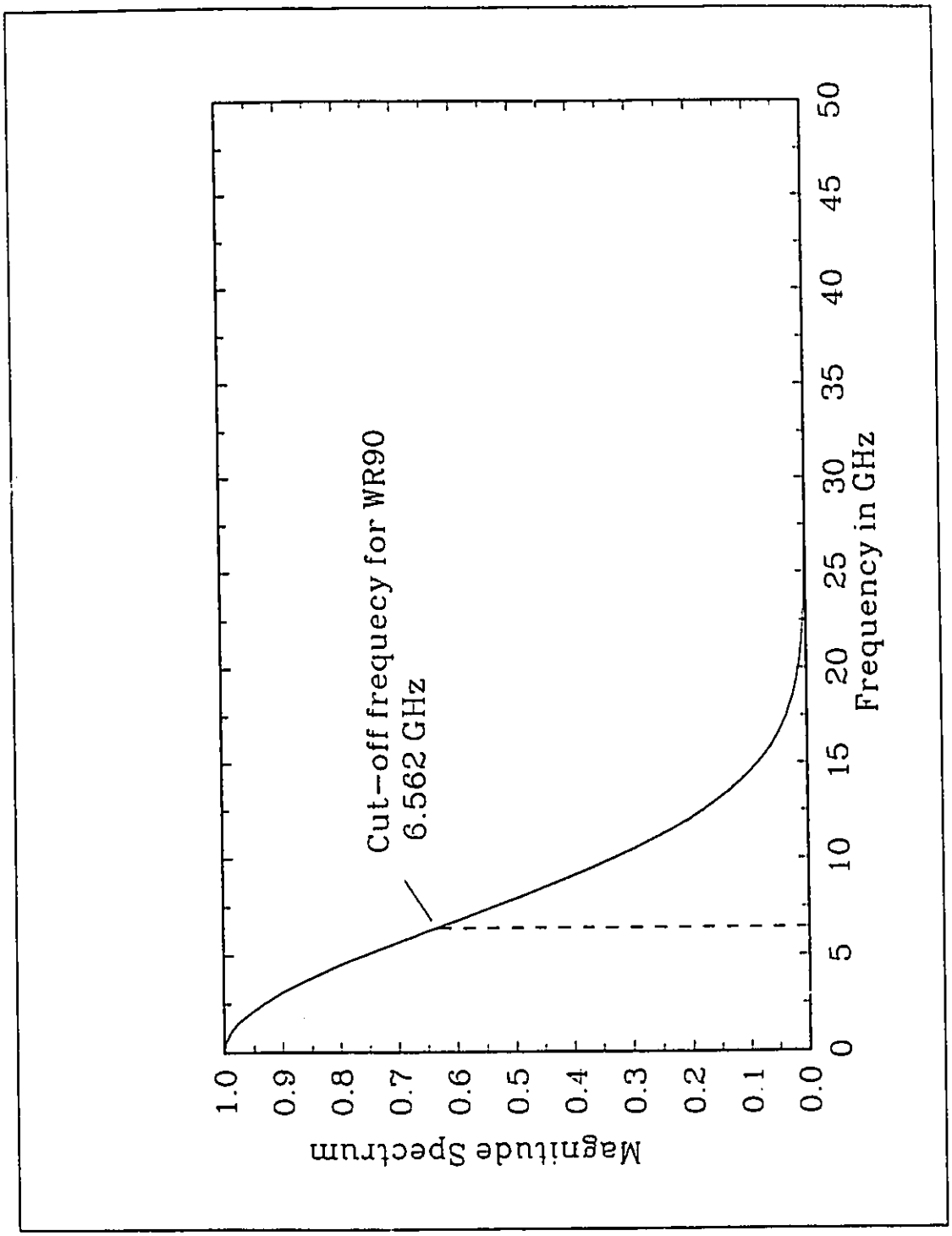


Figure 3.6: Magnitude spectrum of excitation functions: equations 3.1 or 3.2

mode in the cross-sectional plane j as a function of time t. In order to generate the standing-wave pattern, the frequency content of $A^{TE10}(j,t)$ must be determined. The Fourier transform then becomes:

$$A^{TE10}(j,\omega) = \int_{-\infty}^{\infty} A^{TE10}(j,t)\mathrm{e}^{-j\omega t}dt \tag{3.4}$$

Equation 3.4 is the continuous time expression of the Fourier transform for this given problem. In discrete time, the integral becomes a summation over all the time steps (1000). The continuous time variable t becomes $n\Delta t$ (i.e. integer multiples of the discrete time step interval Δt), dt becomes Δt and $A^{TE10}(j,t)$ becomes ${}_nA^{TE10}(j)$. Equation 3.4 can be re-written as follows:

$$\begin{aligned}
A^{TE10}(j,\omega) &= \sum_{n=1}^{1000} {}_nA^{TE10}(j)\mathrm{e}^{-j\omega n\Delta t}\Delta t \\
A^{TE10}(j,f) &= \Delta t \sum_{n=1}^{1000} {}_nA^{TE10}(j)\mathrm{e}^{-j2\pi f n\Delta t} \\
&= \Delta t\left[\sum_{n=1}^{1000} {}_nA^{TE10}(j)\cos\left\{2\pi n\Delta\frac{f}{c}\right\}\right. \\
&\quad \left. - j\sum_{n=1}^{1000} {}_nA^{TE10}(j)\sin\left\{2\pi n\Delta\frac{f}{c}\right\}\right]
\end{aligned} \tag{3.5}$$

Recall that $\omega = 2\pi f$, $\Delta t = \frac{\Delta}{c}$ and $\mathrm{e}^{-jx} = \cos(x) - j\sin(x)$. Note that equation 3.5 represents a complex quantity. When j appears within parentheses it is an index, otherwise it represents the complex operator $j = \sqrt{-1}$.

It remains to determine ${}_nA^{TE10}(j)$, the weighting coefficient for the TE_{10} mode in the cross-sectional plane j at the time step n. In [27, page 172, equation (4-73)], Harrington provides an equation to calculate the weighting (he refers to them as Fourier) coefficients as follows:

$$A^{TE10}(y = y_0, t) = \frac{2}{ab}\int_0^a \int_0^b E_z(x, y = y_0, z, t)\sin\left\{\frac{\pi x}{a}\right\} dzdx \tag{3.6}$$

Where a and b are the dimensions of the waveguide's cross-section, 23Δ and 10Δ respectively, and $E_z(x, y = y_0, z, t)$ represents the time-dependent variation of the E_z field in a given cross-sectional plane, $y = y_0$. For discrete space and time, the cross-section is as shown in Figure 3.7. The integral operators of equation 3.6 become summations and dz and dx become Δ. The whole equation can be re-written as

follows:

$$
\begin{aligned}
{}_nA^{TE10}(j=20) &= \frac{2}{230\Delta^2}\sum_{i=1}^{23}\sum_{k=1}^{10} {}_nE_z(i,j=20,k)\sin\left\{\frac{\pi(i-\frac{1}{2})\Delta}{23\Delta}\right\}\Delta^2 \\
&= \frac{2}{230}\sum_{i=1}^{23}\sum_{k=1}^{10} {}_nE_z(i,j=20,k)\sin\left\{\frac{\pi(i-\frac{1}{2})}{23}\right\} \qquad (3.7)
\end{aligned}
$$

Note that in equation 3.7, ${}_nE_z(i,j=20,k)$ represents the E_z node field at each node in the cross-section j=20 from all possible modes, not just TE_{10}.

In chapter 2.4, it was decided that free space would be characterized by $\mu_r = \epsilon_r = 2$. This entails that, in order for EM propagation to be correctly modelled, any dielectric material in the simulation must use an ϵ_r and μ_r twice their real value (as was done in chapter 2.5). This artificiality must be corrected when considering comparison of simulation data to physical measurements.

The magnitude of $A^{TE10}(j,f)$, or more simply, $\left|A^{TE10}(j,f)\right|$, is formed by taking the square root of the sum of the squares of the real and imaginary parts of $A^{TE10}(j,f)$ as expressed in equation 3.5. This, however, assumes a free space environment where $\mu_r = \epsilon_r = 2$. In order to correct $\left|A^{TE10}(j,f)\right|$ for true free space (i.e. $\mu_r = \epsilon_r = 1$), simply multiply $\left|A^{TE10}(j,f)\right|$ by $\sqrt{2}$ and double the frequency scale. Corrected values for $\left|A^{TE10}(j,f)\right|$ shall be denoted as $\left|A^{TE10}(j,f)\right|^c$.

The reader will observe that, for this method of calculating Γ, the multiplying effect of Δt in equation 3.5 and $\sqrt{2}$ for correction to true free space are nullified by equation 3.8.

In order to generate the standing-wave pattern one must first select the frequency of interest, say f_o. Then, a cross-sectional plane is chosen, say j=20. At each time step n, evaluate ${}_nA^{TE10}(j=20)$ with equation 3.7. Once the simulation is complete, evaluate $\left|A^{TE10}(j=20,f=f_o)\right|^c$ (i.e. the corrected magnitude of $A^{TE10}(j=20,f=f_o)$) with equation 3.5. The result will be the first point in the standing-wave pattern. To obtain the complete pattern one must vary j over the whole range from j=20 to j=70 as shown in Figure 3.8. The result is the standing-wave pattern for the frequency f_o; other standing-wave patterns can be generated at different frequencies by varying f.

One can now determine the voltage standing-wave ratio ($VSWR$) by noting the maximum and minimum values in the standing-wave pattern (as can be seen in Figure

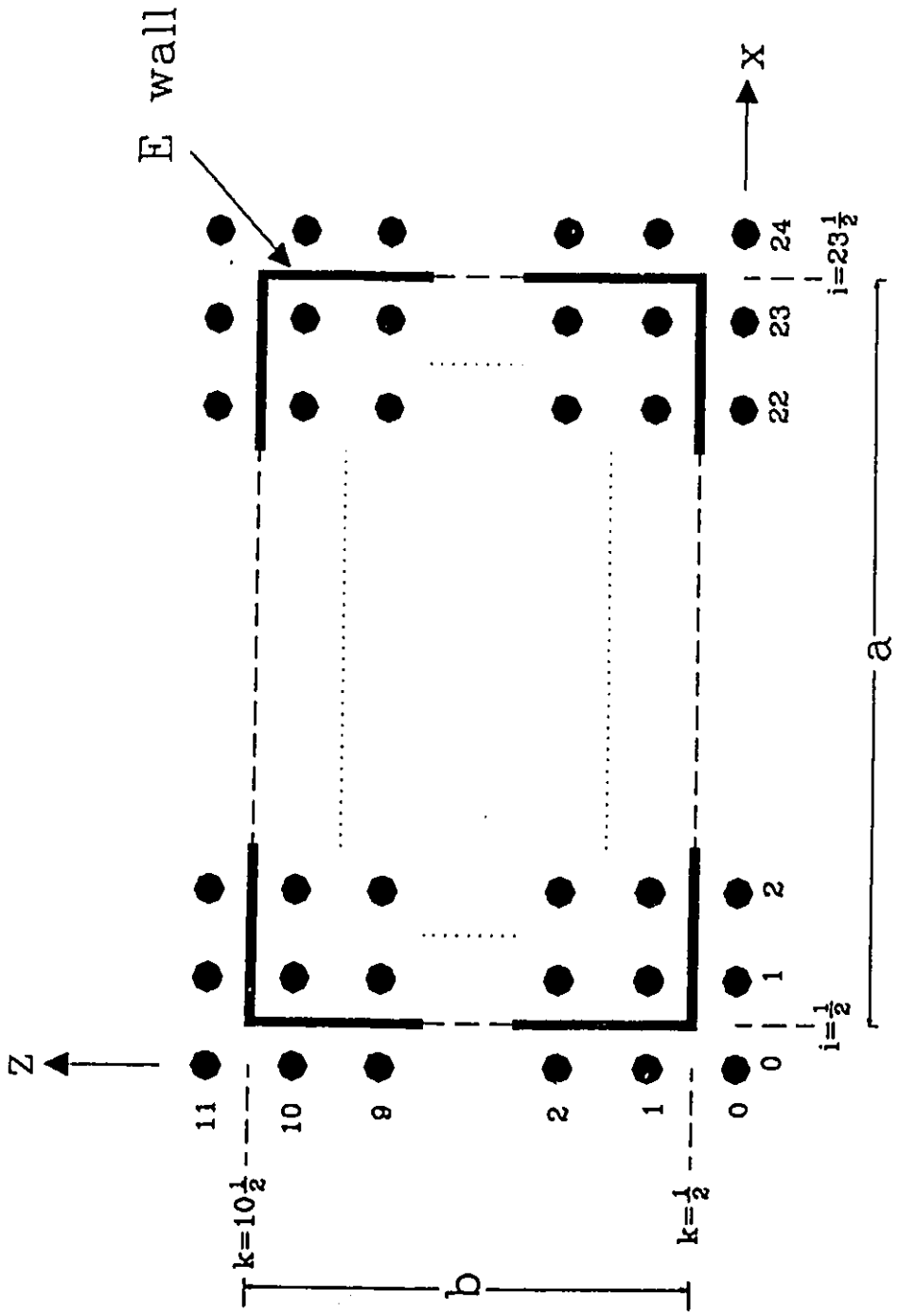


Figure 3.7: Cross-section of the waveguide at j=20

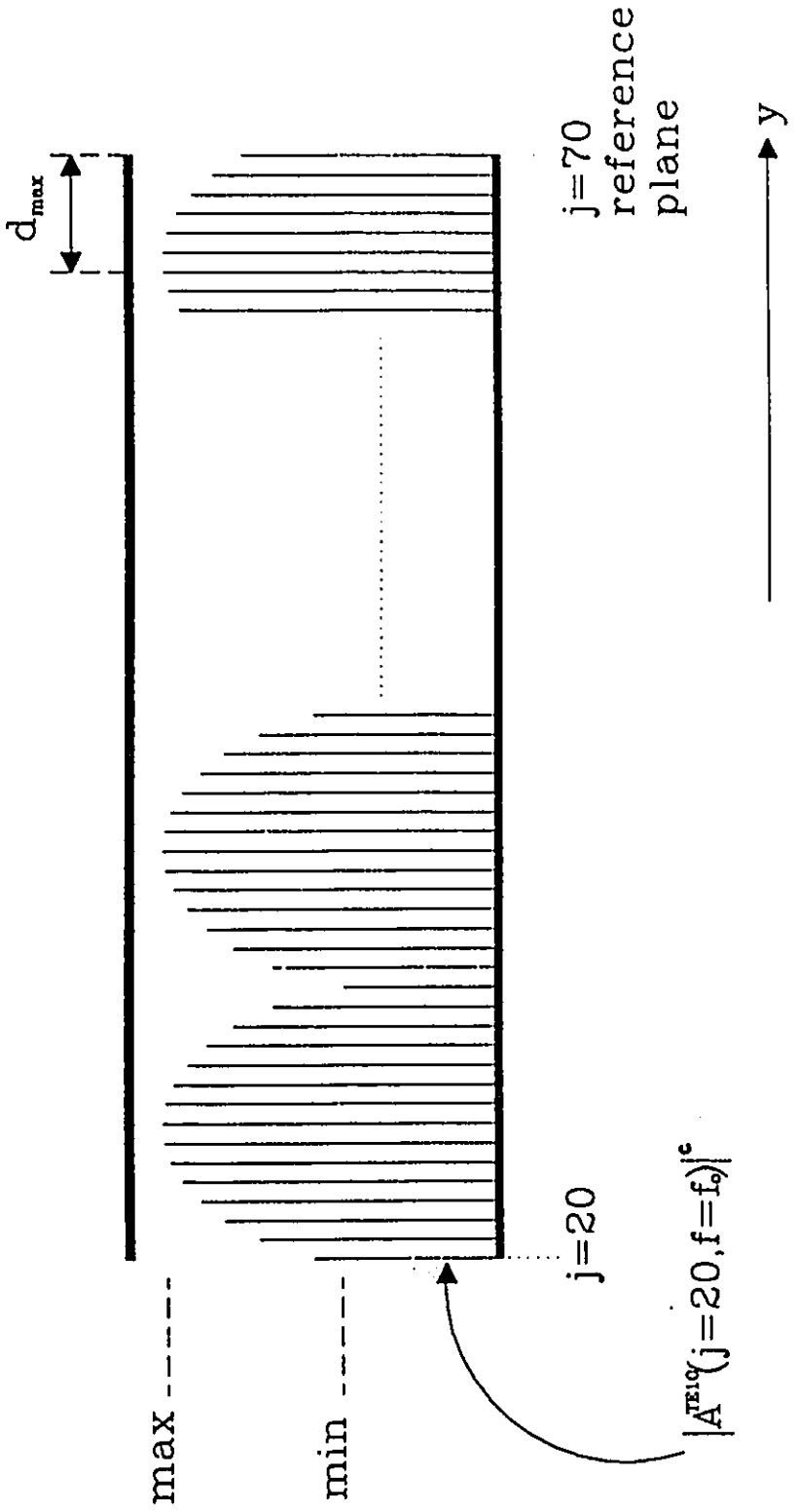


Figure 3.8: Standing-wave pattern along length of the waveguide

3.9) of a particular frequency.

$$VSWR(f) = \frac{\left|A^{TE10}(j,f)\right|^{c}_{max}}{\left|A^{TE10}(j,f)\right|^{c}_{min}} \tag{3.8}$$

The magnitude of the reflection coefficient ($|\Gamma(f)|$) for a given frequency is then given by:

$$|\Gamma(f)| = \frac{VSWR(f) - 1}{VSWR(f) + 1} \tag{3.9}$$

The phase of the reflection coefficient ($phase\Gamma(f)$) at a given frequency is determined by the distance from the reference plane (j=70) to the first maximum in the standing-wave pattern (d_{max}).

$$phase\ \Gamma(f) = \frac{4\pi d_{max}}{\lambda_g} \tag{3.10}$$

Where λ_g is the guided wavelength given by:

$$\begin{aligned} \lambda_g &= \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \\ \lambda_c &= 2a \\ \lambda &= \frac{c}{f} \end{aligned}$$

λ_c is the cut-off wavelength of the guide.

3.4 Measured and Simulated Values for Γ

Values for Γ from the simulation and from measurements will be presented in this chapter. Values for $|\Gamma|$ and $phase\Gamma$ will be compared separately for each of the three EXPAR structures.

Figure 3.9 compares $|\Gamma|$ while Figure 3.10 compares $phase\Gamma$ from simulation and measurements for the EXPAR1 structure. Similarly, Figures 3.11 and 3.12 compare $|\Gamma|$ and $phase\Gamma$ respectively for the EXPAR2 structure; finally, Figures 3.13 and 3.14 show the same comparison for the EXPAR3 structure.

With the exception of some variation at the lower frequencies, the simulated values for $phase\Gamma$ agreed fairly well with the measurements for all three structures as can be seen by Figures 3.10, 3.12 and 3.14. However, $|\Gamma|$ reported by the simulation was less well behaved and showed greater variation when compared to measured values. Since good agreement was achieved with $phase\Gamma$, it need not be considered further.

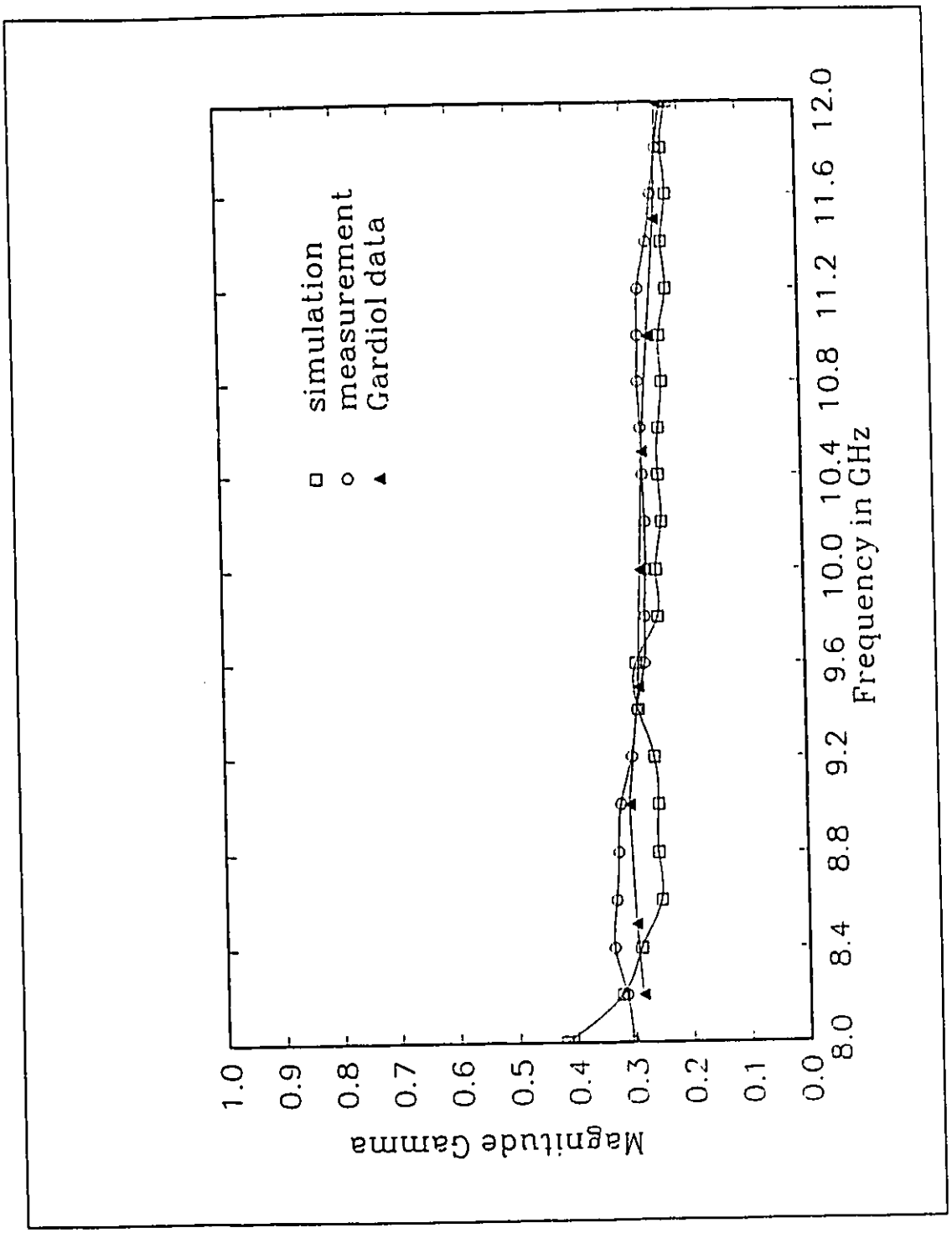


Figure 3.9: $|\Gamma|$ for EXPAR1 structure and from Gardiol's data

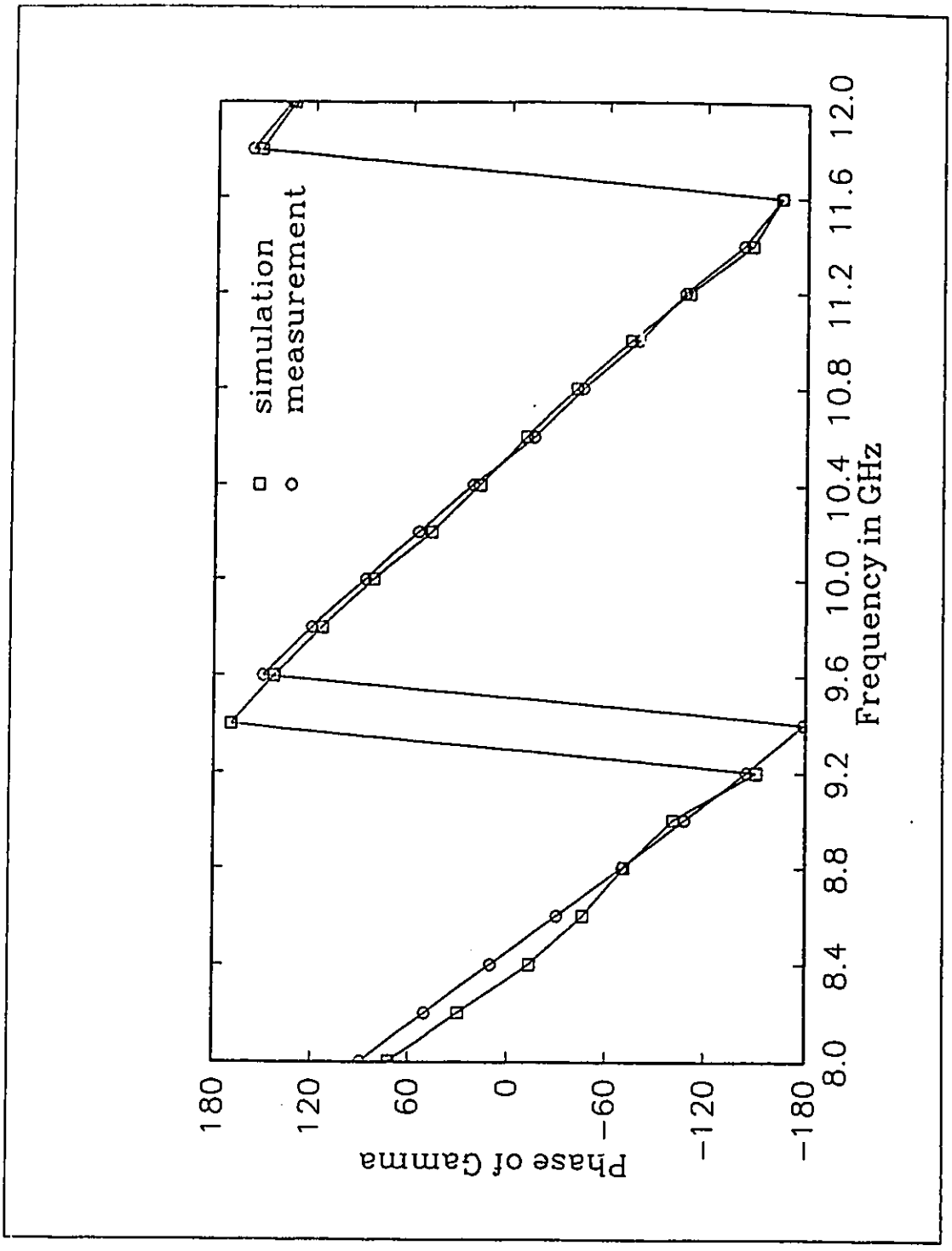


Figure 3.10: *phase*Γ for EXPAR1 structure

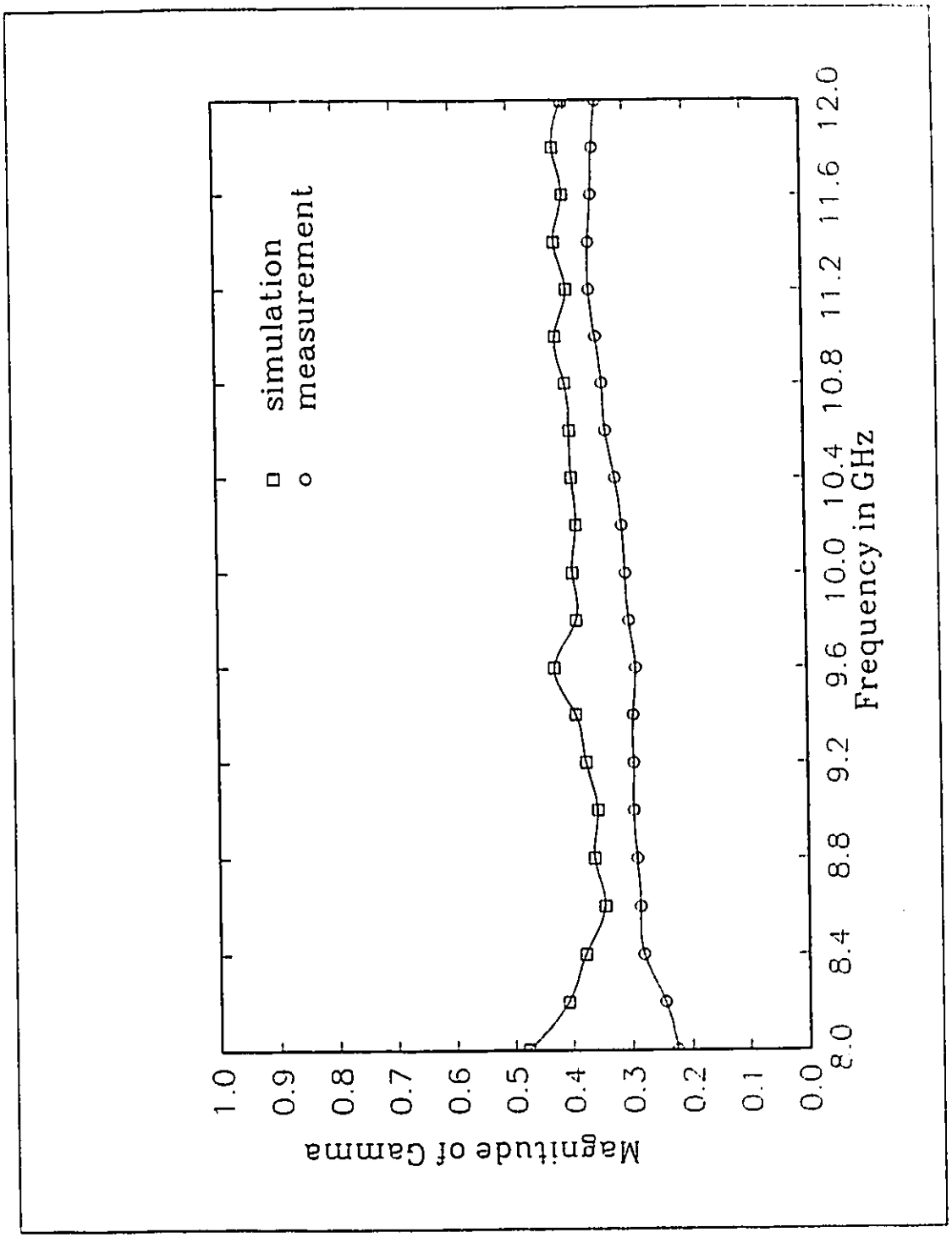


Figure 3.11: $|\Gamma|$ for EXPAR2 structure

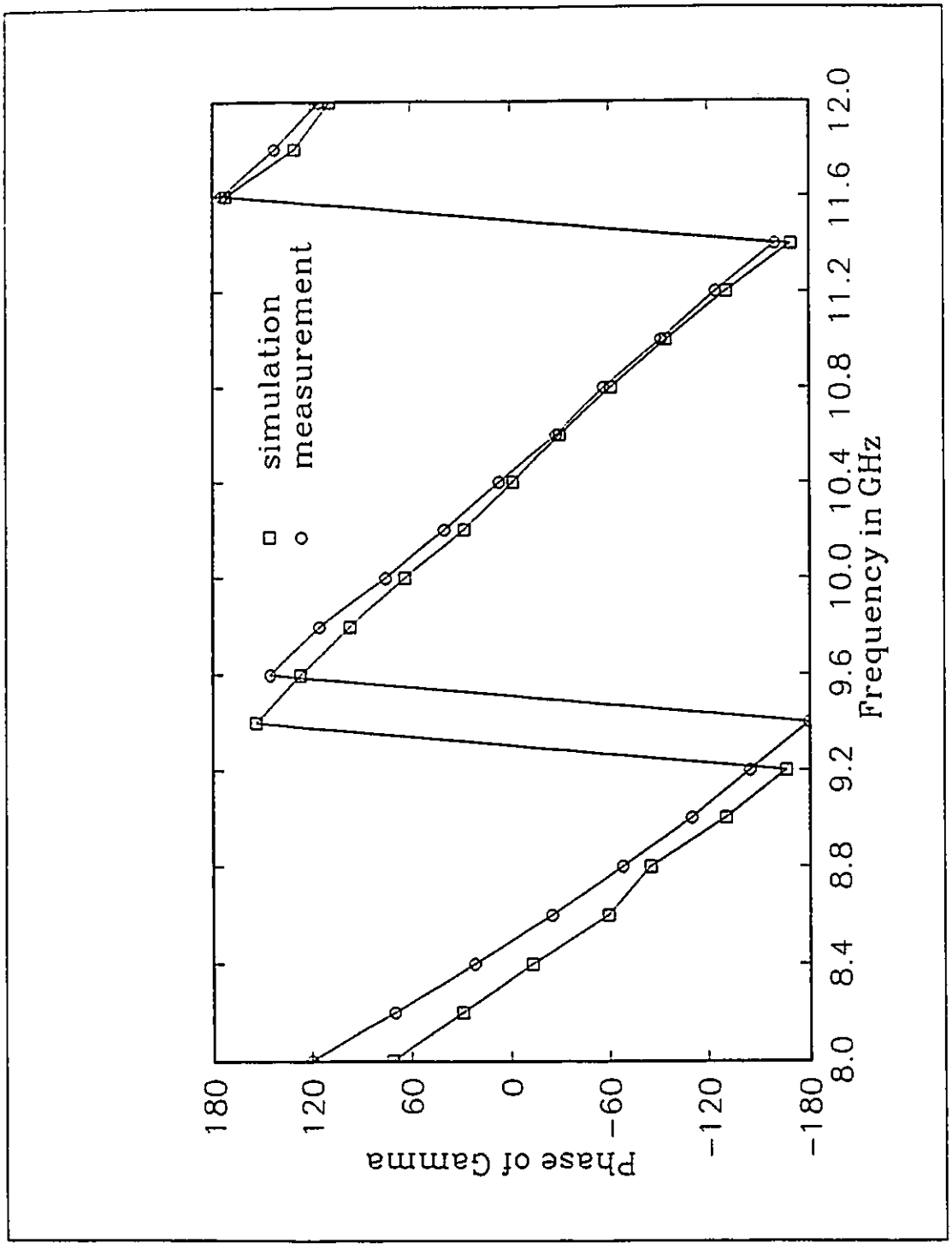


Figure 3.12: *phase*Γ for EXPAR2 structure

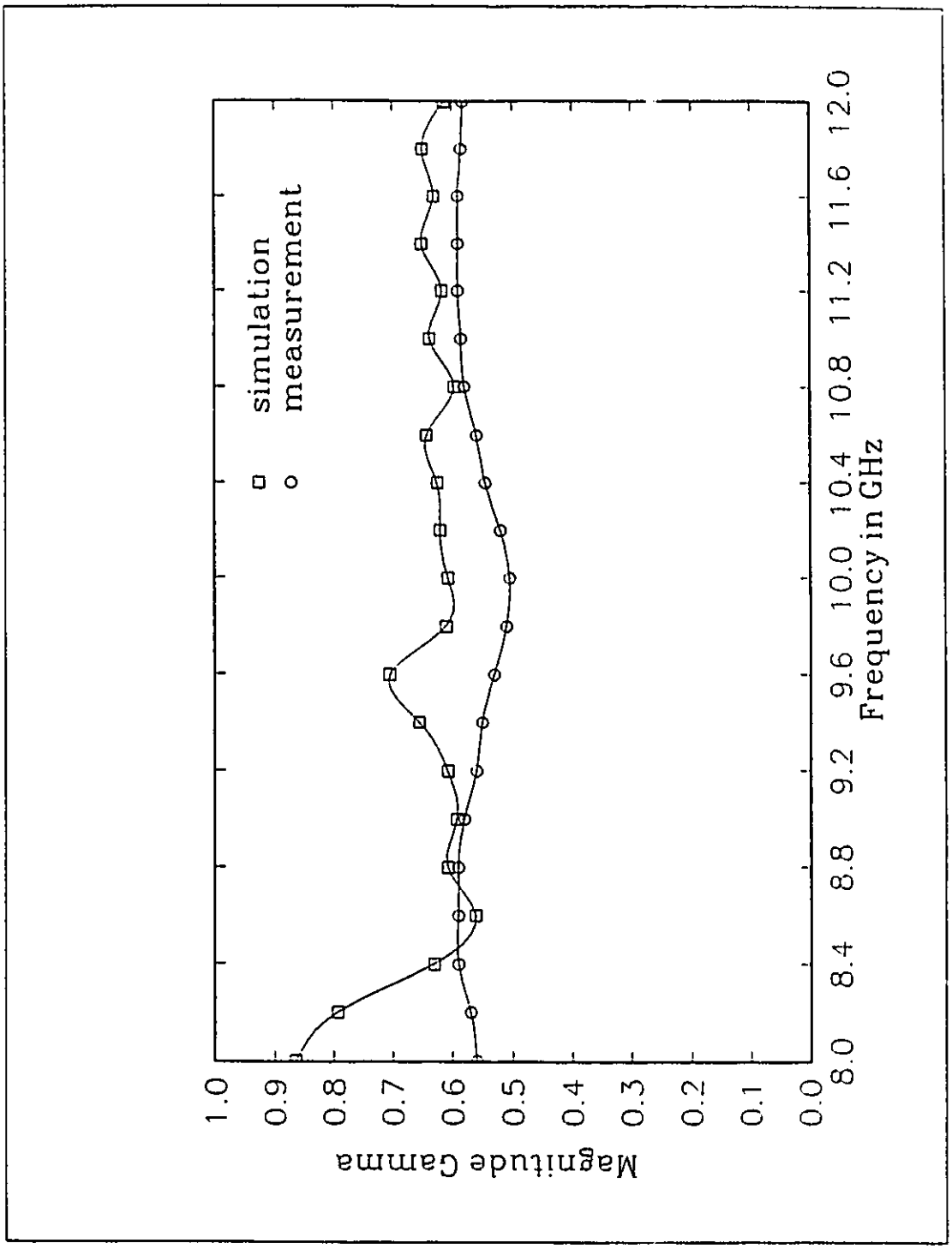


Figure 3.13: $|\Gamma|$ for EXPAR3 structure

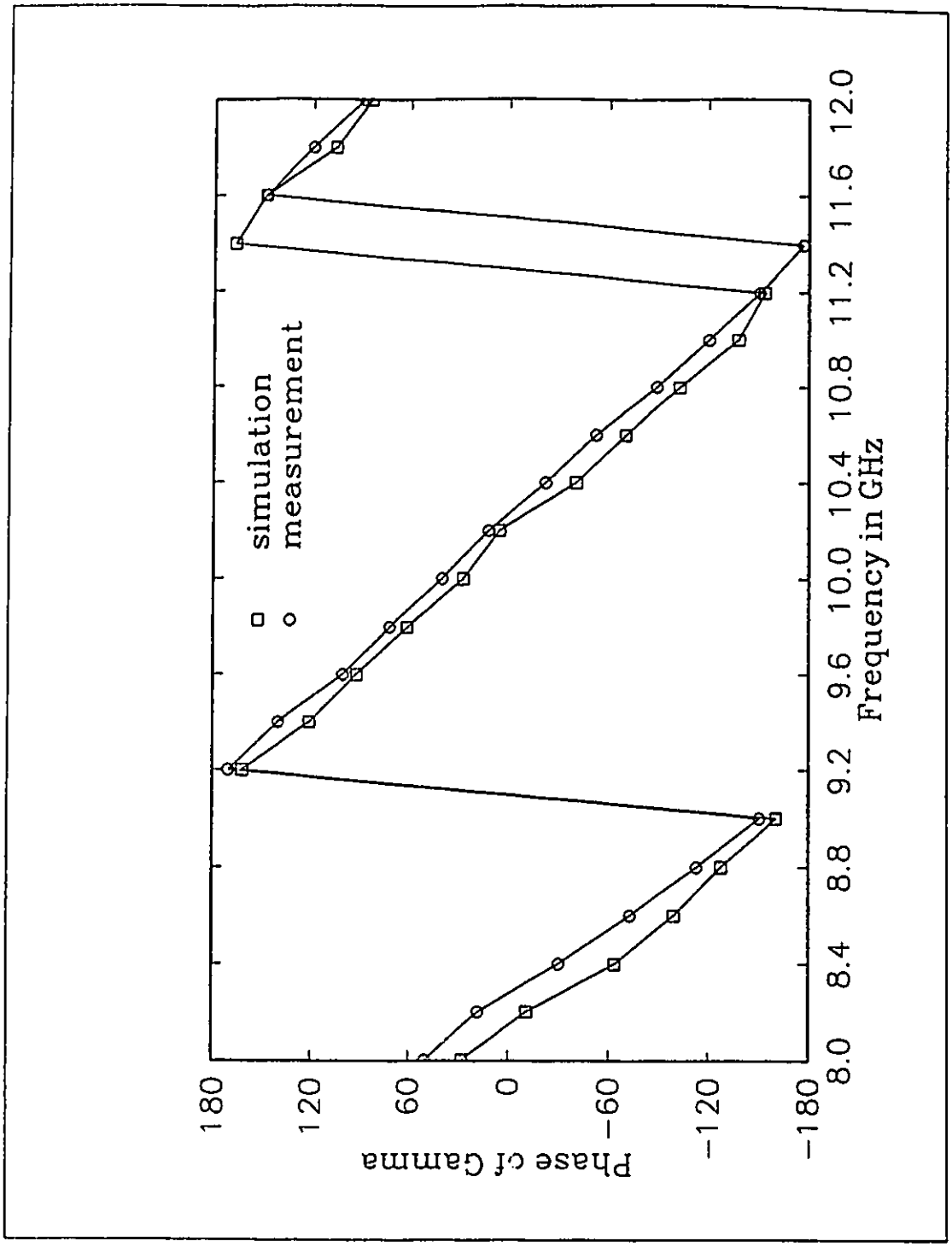


Figure 3.14: *phase*Γ for EXPAR3 structure

The rest of this chapter will examine sources of error to which the discrepancies, as observed in Figures 3.9, 3.11 and 3.13, may be attributed.

Before discussing sources of error, the reader should pay particular attention to Figure 3.9. In addition to $|\Gamma|$ data from the simulation and measurement, this figure also shows data from a paper by Gardiol [25] who conducted a similar measurement. Gardiol measured the $VSWR$ of an open-ended X-band rectangular waveguide (which is exactly EXPAR1 in this study) and his results are presented in Figure 3.9 (the $VSWR$ data is converted to $|\Gamma|$ by equation 3.9). Gardiol's measurements agree with those done by the author. It is hoped that, by emphasizing this agreement the reader will feel greater confidence in the measurements shown in subsequent figures.

The first and most obvious possible source of error to be discussed is the influence of the waveguide flange upon $|\Gamma|$ measurements. The flange is used to connect the physical EXPAR structures (see Figure 3.1) to the measurement apparatus. Since it was not modelled, it is reasonable to assume that it may be the cause of the discrepancies noted above.

The waveguides used in the measurements and simulations are lossless. It is known that, for lossless waveguides, $|\Gamma|$ is constant throughout the length of the guide. It was hypothesized that, for EXPAR1, if the flange was influencing the $|\Gamma|$ measurements, then extending the structure (i.e. putting more waveguide between the flange and the aperture) should alter the measurements because the effect of the flange would decrease as the distance from flange to aperture increases.

The physical EXPAR1 device is 5 cm long. A structure, identical in all respects to EXPAR1 except that its length was 10 cm, was built. Let this new structure be called EXPAR1*. $|\Gamma|$ for EXPAR1* was measured and compared to that for EXPAR1 as shown in Figure 3.15. From this diagram no appreciable variation in $|\Gamma|$ for the longer structure, EXPAR1*, was observed.

It must be concluded that the flange of the three EXPAR structures is already far enough away from the aperture so as to have no significant effect upon $|\Gamma|$ measurements. Therefore, the author concludes that the presence of waveguide flange does not account for the discrepancies observed in Figures 3.9, 3.11 and 3.13.

Another possible source of error is that which is inherent to all numerical simulations: round-off error. In order to perform the simulation, the dimensions of the waveguide were rounded off from 10.16×22.86 mm to 10×23 mm and the thickness of the dielectric covering was assumed to be 2 mm as opposed to 1.6 mm (for the

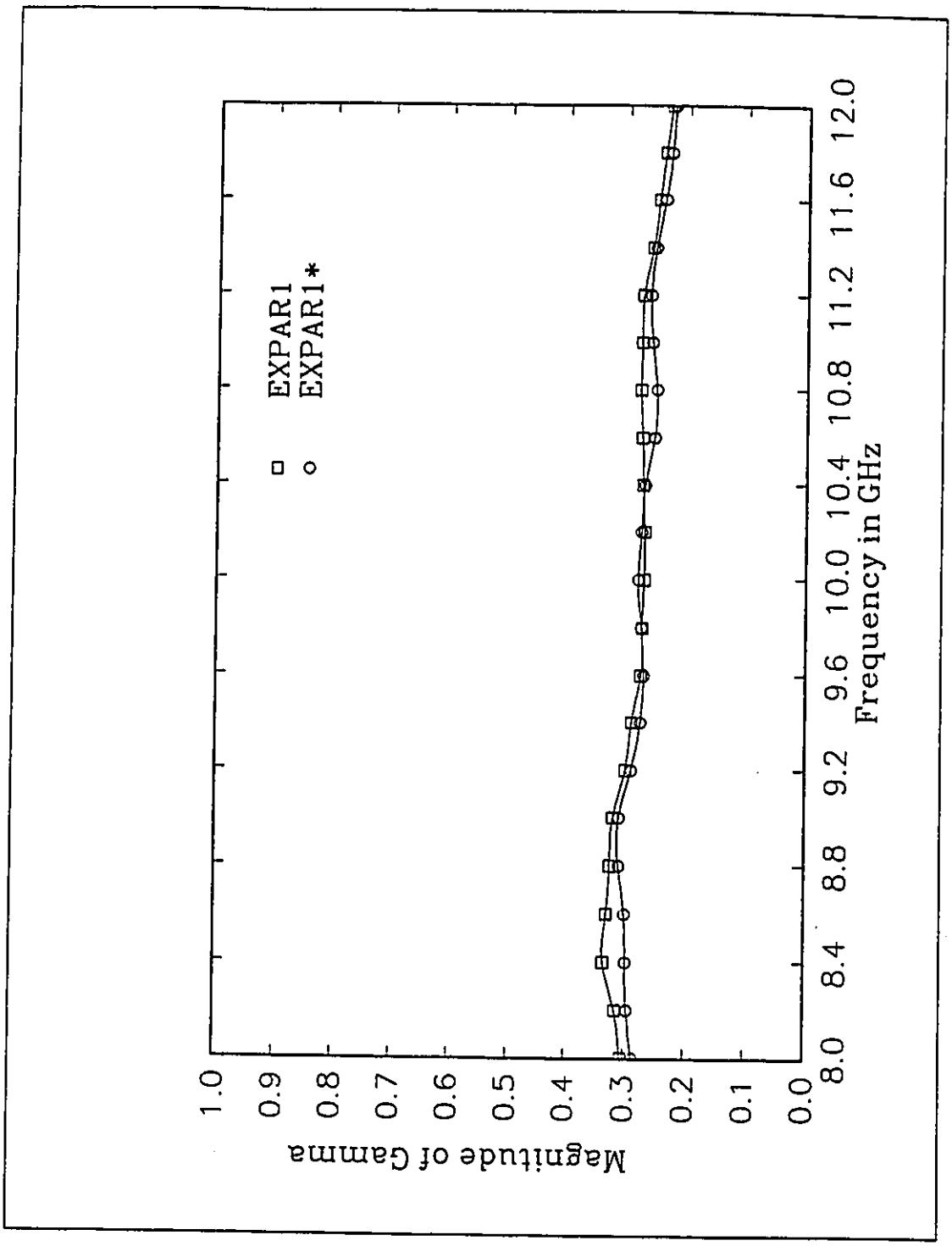


Figure 3.15: Measured values of $|\Gamma|$ for EXPAR1 and EXPAR1* structures

EXPAR3 structure only). Over the frequency range of interest (8 to 12 GHz), the shortest wavelength is 25 mm. The amount of round off compared to this wavelength yields the maximum round-off error experienced over the whole frequency range. The errors for the waveguide dimensions are 0.64% and 3.4% for the height and width respectively. The error for the dielectric's thickness is 1.6%.

Bearing in mind that these are the maximum round-off errors, one can conclude that the approximations made should not significantly affect the simulation results. Consequently, the round-off errors, by themselves, cannot account for the discrepancies observed in Figures 3.9, 3.11 and 3.13.

The relative dielectric constant (ϵ_r) of the covering is 2.33±.02@10 GHz as already mentioned in chapter 3.1. It is possible that this uncertainty contributed to the discrepancy observed between measured and simulated values for $|\Gamma|$ for the EXPAR3 structure (Figure 3.14).

In order to investigate the effect of the ϵ_r uncertainty upon $|\Gamma|$, two additional simulations were performed with EXPAR3: one with ϵ_r=2.35 and another with ϵ_r=2.31. The results were then compared with the original simulation (i.e. ϵ_r=2.33). It was found that, with ϵ_r=2.35, $|\Gamma|$ was, for the most part, 0.5% greater; with ϵ_r=2.31, $|\Gamma|$ was about 0.5% less.

The inescapable conclusion is that the uncertainty associated with ϵ_r did not contribute significantly to the discrepancy observed in Figure 3.13. Since the structures EXPAR1 and EXPAR2 did not contain any dielectric material, this analysis does not apply to Figures 3.9 or 3.11.

Yet another possible source of error is that of unwanted reflections from the AWs in the free space region (FSR). This arises because, perhaps, not all the energy incident upon the AWs is absorbed, some may be reflected due to the proximity of the AWs. One may suppose that the AWs are too close to the energy source and, as a result, do not effectively absorb incident radiation (and hence affect $|\Gamma|$). If they were placed farther away, they might absorb better.

In order to examine any effect the AWs may have had upon the simulation, it was decided to alter the position of one of the AWs (and consequently the size of the FSR) for EXPAR3 and note any variation in $|\Gamma|$ data. Only the distance between the aperture and the far AW along the y axis (i.e. the AW in the xz plane) was varied. This course of action was chosen for two reasons: It is simpler to move one AW than all six of the FSR. Selecting the far AW along the y axis is logical because this AW

receives most of the incident energy radiating from the aperture; therefore, of all six AWs in the FSR, it should have the greatest effect (if any) upon the simulation.

The standard EXPAR3 structure (i.e. Figure 3.3(c)) has the far AW located 33 mm (or 33Δ) from the aperture. The far AW was moved in to 24 mm and then moved out to 44, 54 and 74 mm. At each position, $|\Gamma|$ data was compared to the standard. For all positions, no significant change in $|\Gamma|$ was observed. (In fact, in most cases, there was no change even to third decimal place!)

One must conclude that the current location of the far AW had no significant impact whatsoever upon the simulation. The AWs may yet have an effect upon the simulation; perhaps if they were flush to the aperture. The point here is that the above mentioned results suggests that their current position does not influence the simulation; therefore, the discrepancies observed in Figures 3.9, 3.11 and 3.13 are not due to their location .

The author is left to conclude that the manner in which $|\Gamma|$ is calculated is itself the source for the discrepancies that have been observed. The method for determining $|\Gamma|$ was discussed in chapter 3.3. It involves generating the standing-wave pattern for frequencies of interest. In [29, page 425], the authors define $VSWR$ as the ratio of the maximum value to an *adjacent* minimum value. This was the approach that was used in this study and is implied by equation 3.8. However, while doing this calculation, the author observed an irregularity: the standing-wave pattern did not exhibit constant maximum or minimum values. A slight variation was noted.

Figures 3.16, 3.17 and 3.18 show simulated standing-wave patterns in the EXPAR3 structure for frequencies of 8, 10 and 12 GHz respectively. The non-uniformity of the maximums and the minimums is more apparent at higher frequencies because there are more cycles of the standing-wave pattern over which a variation can manifest itself.

Due to the variation in the standing-wave pattern, the value of $|\Gamma|$ depended considerably upon which maximum and adjacent minimum was selected. Theoretically, because the structure is lossless, the maximums and minimums should all have the same value throughout the standing-wave pattern. In [15] and [30], the authors noted that the Fourier transform of time-domain data was very sensitive to numerical errors (in the time domain) such as round-off or computer truncation error.

Since generating the standing-wave pattern is essentially a Fourier transform process, the author contends that the variation in the pattern is due to small time-domain

errors (perhaps the round-off error - mentioned earlier in this chapter - is not as innocuous as previously thought). Admittedly, the variation in the pattern is slight but the VSWR (and ultimately $|\Gamma|$) is extremely sensitive to variations in maximum and minimum values because it is the ratio of one to the other. This sensitivity accounts for fluctuations in $|\Gamma|$ as observed in Figures 3.9, 3.11 and 3.13. On the other hand, $phase\Gamma$ depends primarily upon the distance from the reference plane to the first maximum (i.e. d_{max} in equation 3.10) and not upon its value. This helps to account for the better agreement of $phase\Gamma$ data as compared with $|\Gamma|$.

In summary, the author submits that seemingly inconsequential errors in time-domain data, when transformed into the frequency domain, manifested themselves as a slight variation in the standing-wave pattern. Due to the manner in which the $VSWR$ and $|\Gamma|$ are determined, this slight variation is the source of the lack of agreement of the simulated and measured values for $|\Gamma|$.

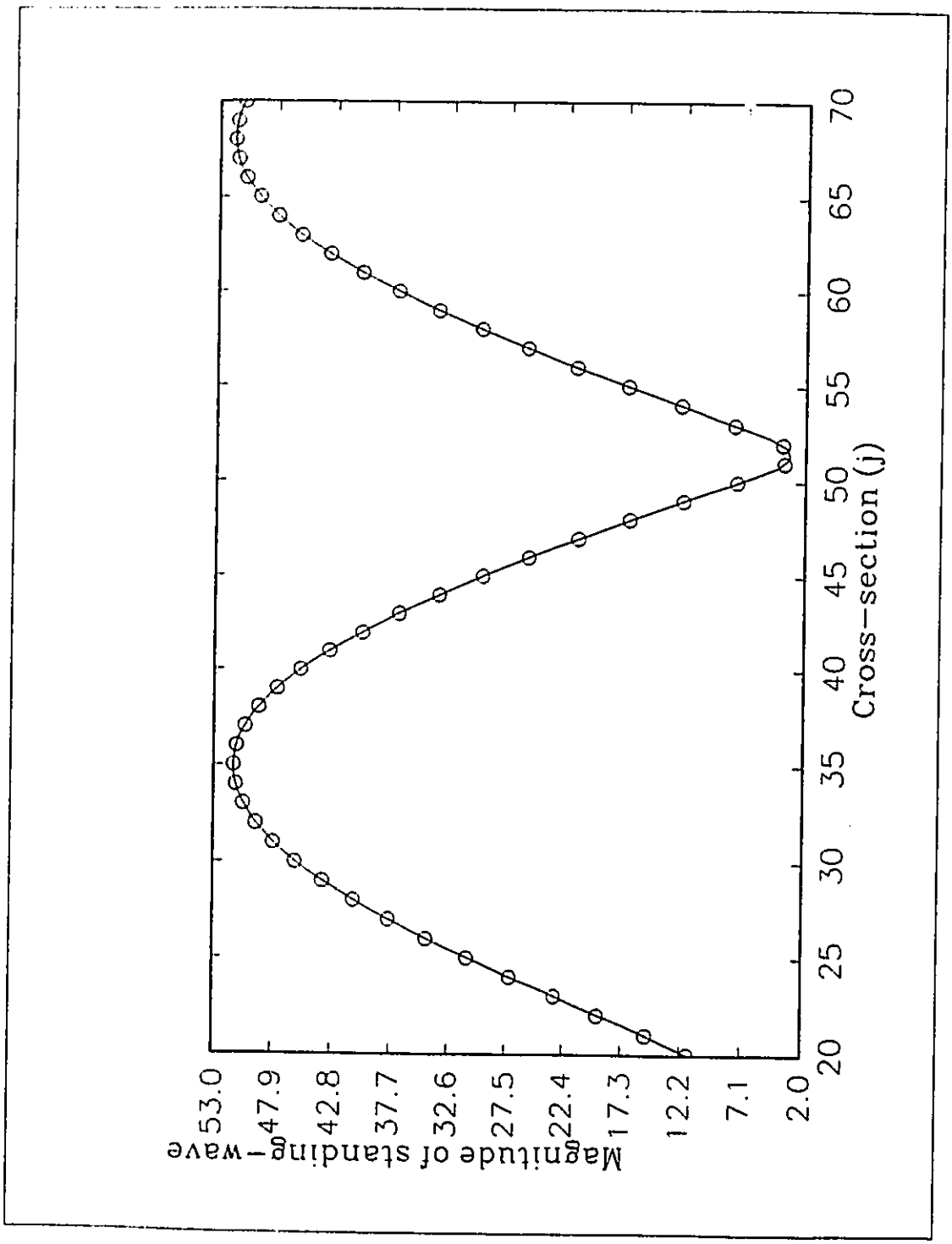


Figure 3.16: Simulated standing-wave pattern for EXPAR3 at 8 GHz

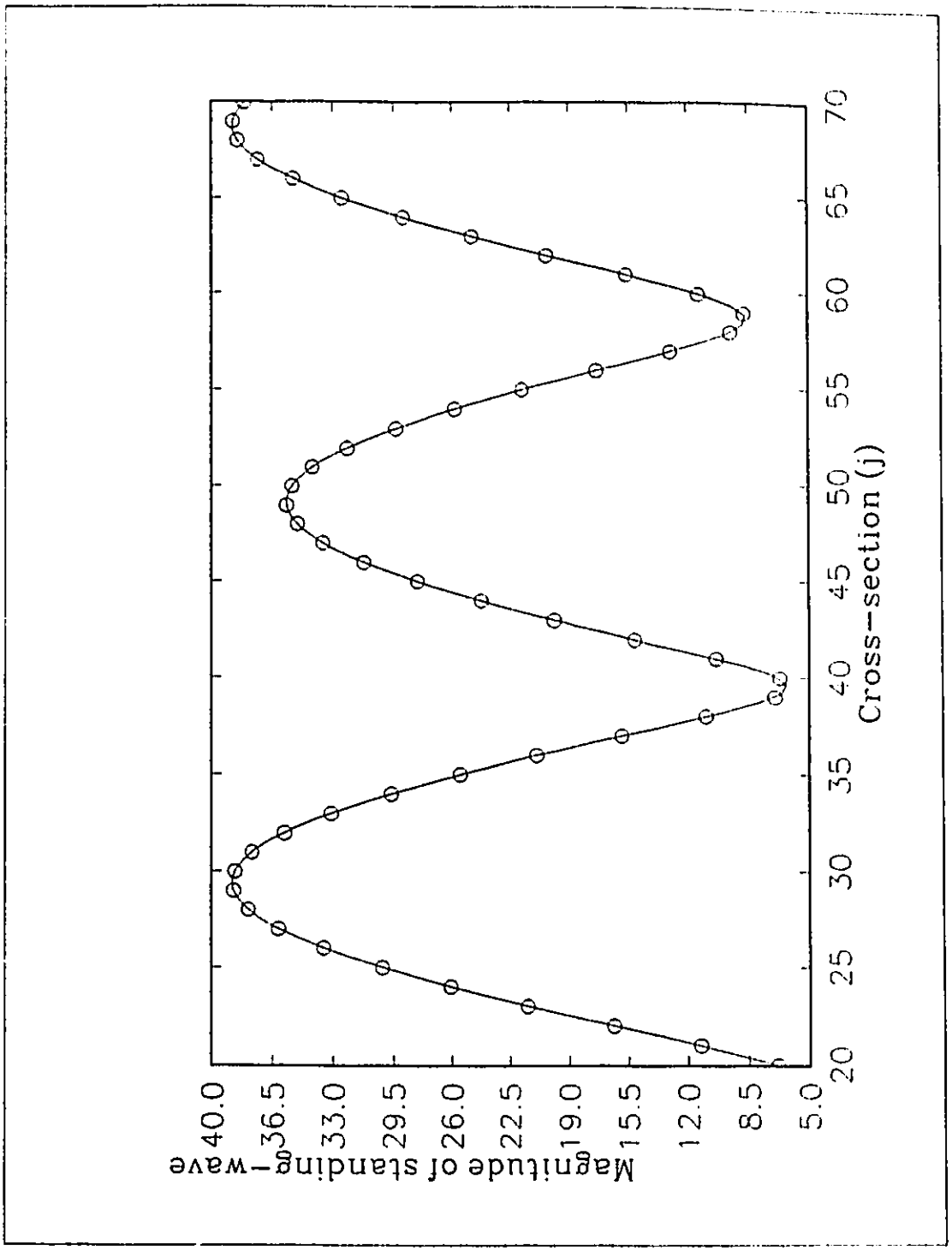


Figure 3.17: Simulated standing-wave pattern for EXPAR3 at 10 GHz

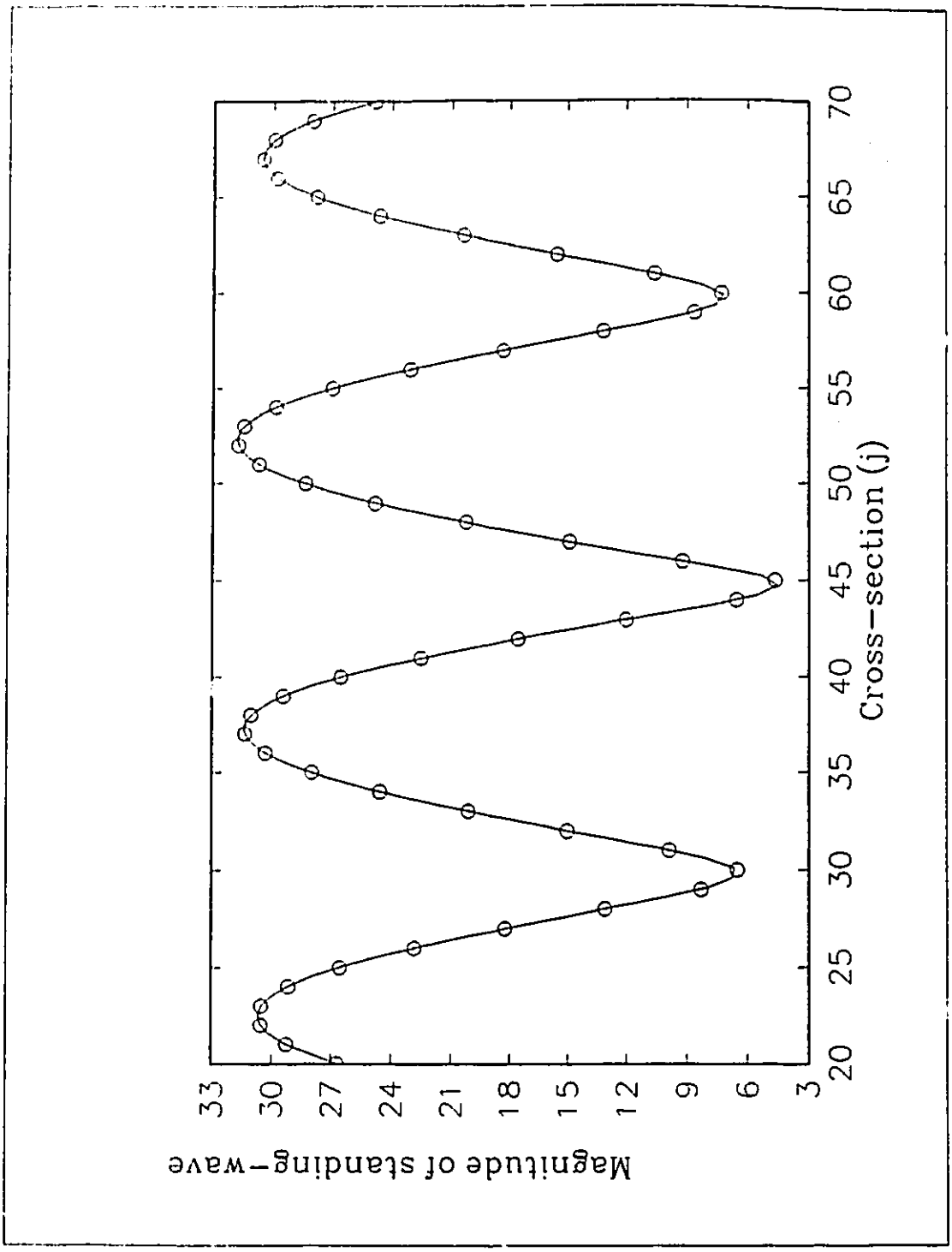


Figure 3.18: Simulated standing-wave pattern for EXPAR3 at 12 GHz

Chapter 4

Simulation Considerations

4.1 Symmetry Considerations

The EM fields are symmetrical about two planes as can be seen from the photographs in the next chapter. The symmetrical nature of the EM fields is a property that can be exploited to reduce the size of structure while maintaining the same performance.

The reader will recall that an E wall is a boundary at which all the electric field is normal (i.e. the tangential E field components are zero) and all the magnetic field is tangential (i.e. the normal magnetic field component is zero) to it. The M wall has the dual properties of the E wall: all magnetic fields are normal and all electric fields are tangential to it. Given the characteristics of these boundaries and photographs of the field distributions in chapter 4.2, it is obvious that an E wall and an M wall could have been placed, as shown in Figure 4.1, without altering the behavior of the EM fields throughout the structure. The reader is reminded that the walls of the waveguide are E wall boundaries but do not apply to this discussion.

Had E and M walls been used, only one quarter of the original structure (be it EXPAR1, EXPAR2 or EXPAR3) would have required simulation in order to obtain the data necessary for $|\Gamma|$ calculations (i.e. the shaded region of Figure 4.1) The question that naturally comes to a critical reader's mind is why, given the limited computer resources, did the author not avail himself of the opportunity to reduce the size of the structure and save valuable CPU time?

There can be no question of the savings offered by the use of E and M walls. However, the author had an important reason for eschewing their use in this case: It was to better convince skeptical readers of the correct performance of the simulation. Any reader can quickly look at the images of the EM field distribution and, by virtue

of symmetry, be confident that the algorithm which produced the data is working correctly.

Without the full image (i.e. the image not truncated by the planes of symmetry), the reader would require prior knowledge of EM field distribution in order to verify correct simulation. This is not difficult for a waveguide with TE_{10} mode propagation but becomes onerous when dealing with higher order modes at discontinuities.

Even though the EM field distribution is symmetrical, it does not guarantee accurate simulation results; there could be other problems. Symmetry of the field does , however, provide the reader with a quick check on performance and suggests that the simulation is operating correctly.

4.2 Information From the Simulation

This chapter will discuss the quantity and quality of the information that can be obtained by the method. Photographs of the simulation at various iterations are provided by way of example.

The amount of data that can be extracted from the simulation is truly voluminous. Using this method, one can theoretically obtain field data for $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ (i.e. all six field components $E_x, E_y, E_z, H_x, H_y, H_z$) at any location (i, j, k) in the problem space for any iteration in time (n). Computer memory however, will ultimately determine how much data an individual user can store.

The following pages in this chapter provide an example of the detailed nature of the information offered by the simulation. Consider the EXPAR1 structure of chapter 3. For the purposes of display, let the waveguide region be shortened from 120 mm to 40 mm (the size of the free space region remains the same) and call this new structure MINI1. In order to investigate EM field behavior in MINI1, consider the viewing planes as shown in Figure 4.2 and 4.3. To view MINI1 from the top, an xy plane cuts MINI1 at k=4 and k=20 in the waveguide and free-space regions respectively (Figure 4.2). The indices in the two regions are dissimilar because the regions that they refer to are of different sizes. To view MINI1 from the side, a yz plane cuts MINI1 at i=13 and i=23 in the waveguide and free space regions respectively (Figure 4.3).

The reader's attention is directed to Figures 4.2 and 4.3 regarding the waveguide walls that extend into the free-space region. The diagrams clearly show a wall thickness of 1Δ. It was unnecessary to model the wall thickness in the waveguide region

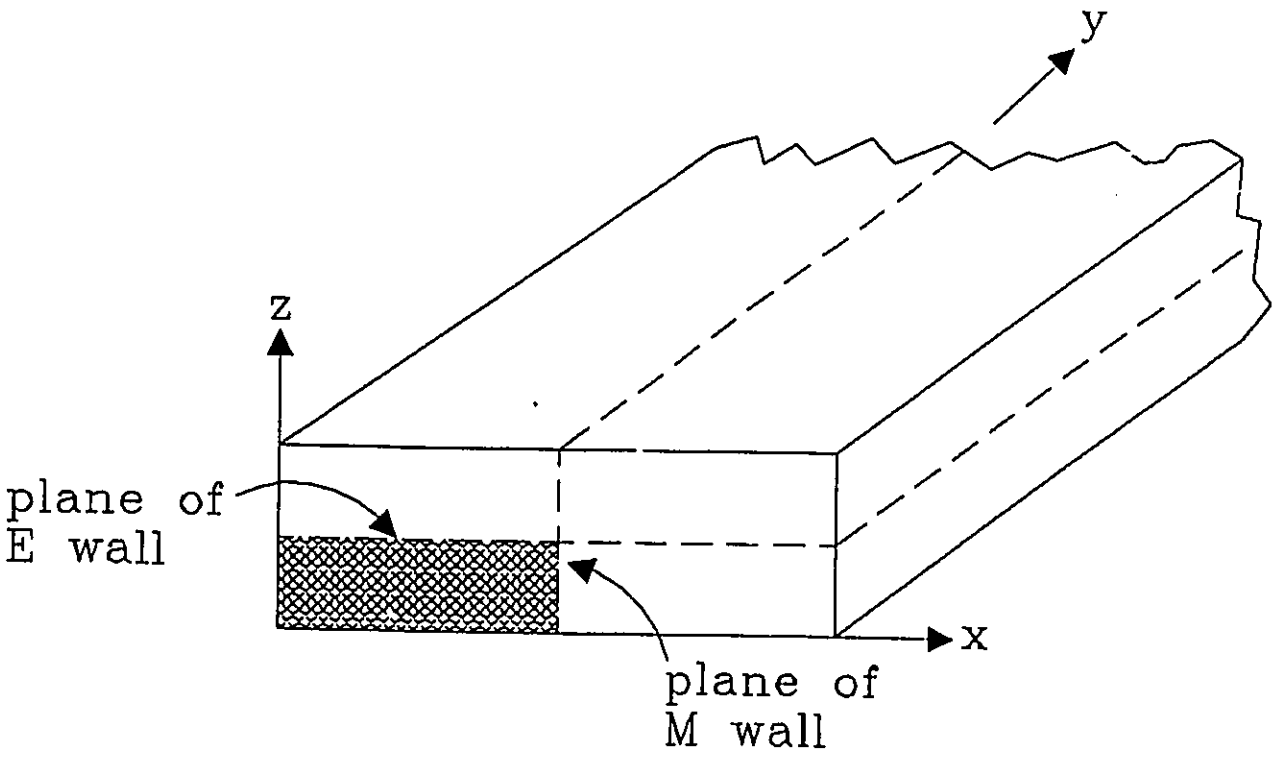


Figure 4.1: Position of E and M walls for symmetry

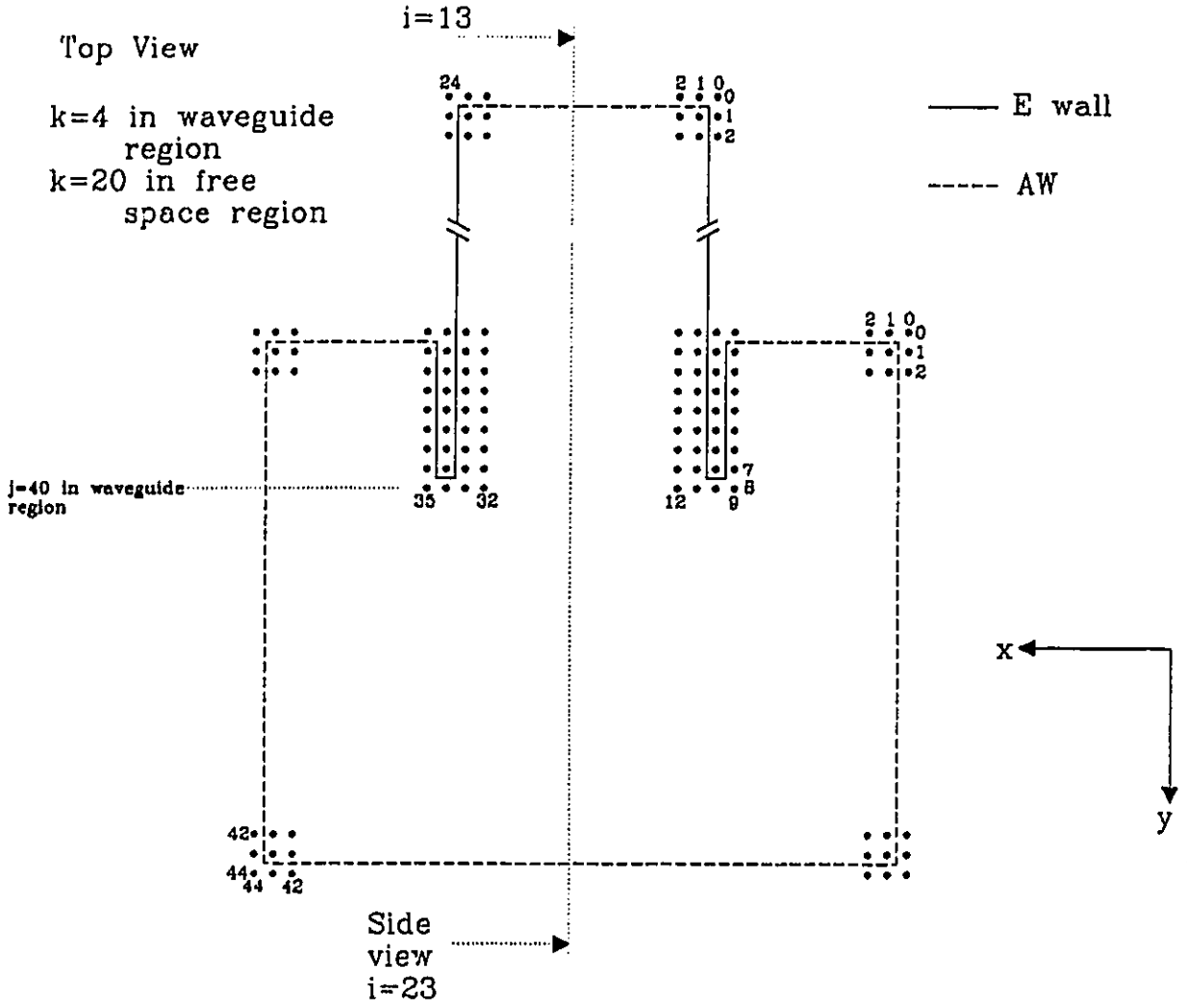


Figure 4.2: Top view of the MINI1 structure

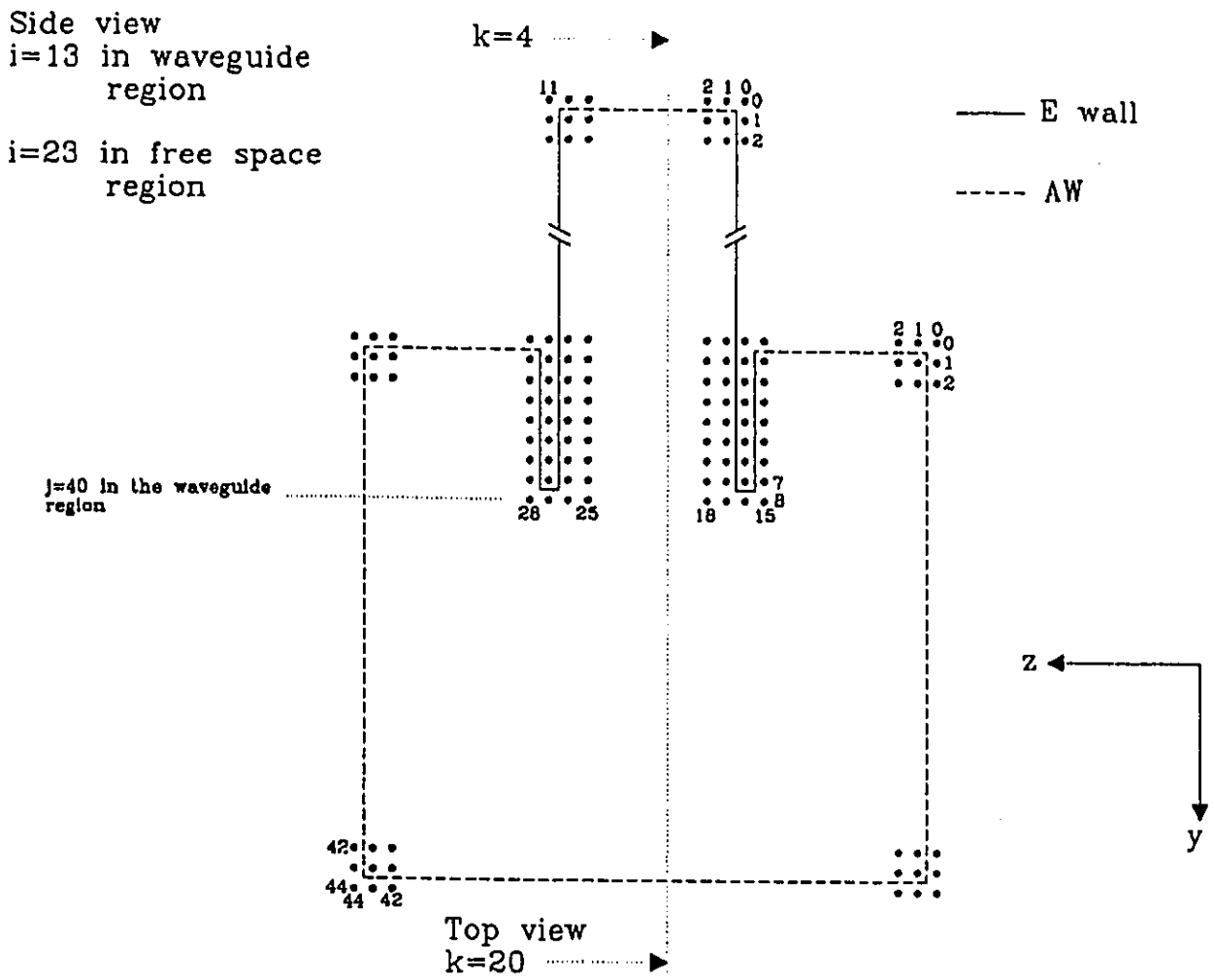


Figure 4.3: Side view of the MIN11 structure

since the E walls there were also the limits to the problem space.

Figures 4.5, 4.6 and 4.7 show the distribution of the E_z field component as seen from the top and side of the MINI structure for iterations 40, 74, 111, 150, 204 and 235. Figure 4.4 also provides a logarithmic colour scale for the magnitude of the field intensity. From these photographs the excitation and propagation of the energy and its symmetrical nature can clearly be seen. Also noteworthy is the curvature of the wave fronts in the free-space region, they do not seem to be distorted suggesting good energy absorption by the AWs.

Figure 4.5 is particularly interesting for two reasons: The photo of iteration 40 shows the excitation at its peak (recall the Gaussian pulse has a mean of 40 - equations 3.1 and 3.2). The photo of iteration 74 shows energy leaving the open end of the waveguide and spilling back along the walls of the guide. From the colour scale one can determine that, for this iteration, the "backward" energy flow is in the order of 20 dB below that which propagates in the forward direction. The salient point here is that the simulation shows energy travelling in a direction that one would least expect.

Figures 4.8, 4.9 and 4.10 show the distribution of the $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ fields as seen from the top of the MINI structure for iterations 45, 76, 92, 120,140 and 150. The $\vec{\mathbf{H}}$ field is multiplied by $Z_o = \sqrt{\frac{\mu_o}{\epsilon_o}} = 377\Omega$ so that its magnitude would be comparable to that of $\vec{\mathbf{E}}$.

Figure 4.8 shows the $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ fields for iterations 45 and 76. In the photo of iteration 45, excitation of the TE_{10} mode is plainly visible (i.e. sinusoidal variation of E_z and H_x, co-sinusoidal variation for H_y and no longitudinal field components present). Energy can also be seen propagating in the opposite direction (also in the TE_{10} mode), i.e. towards the AW not the waveguide aperture as was first discussed in chapter 3.2.

The photo of iteration 76 shows the energy reaching the aperture and, with energy beginning to appear in E_y and H_z, the creation of higher order modes. In subsequent iterations, the effect of the corners of the waveguide walls is very pronounced. These corners act like radiators of energy for the E_x field component.

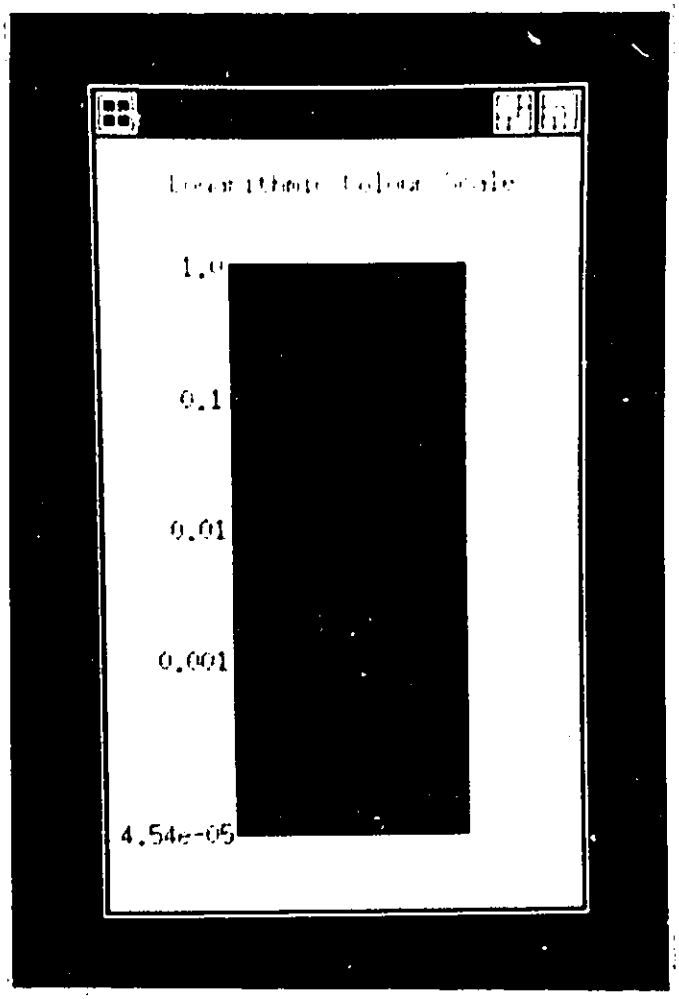


Figure 4.4: Logarithmic colour scale used for all simulations

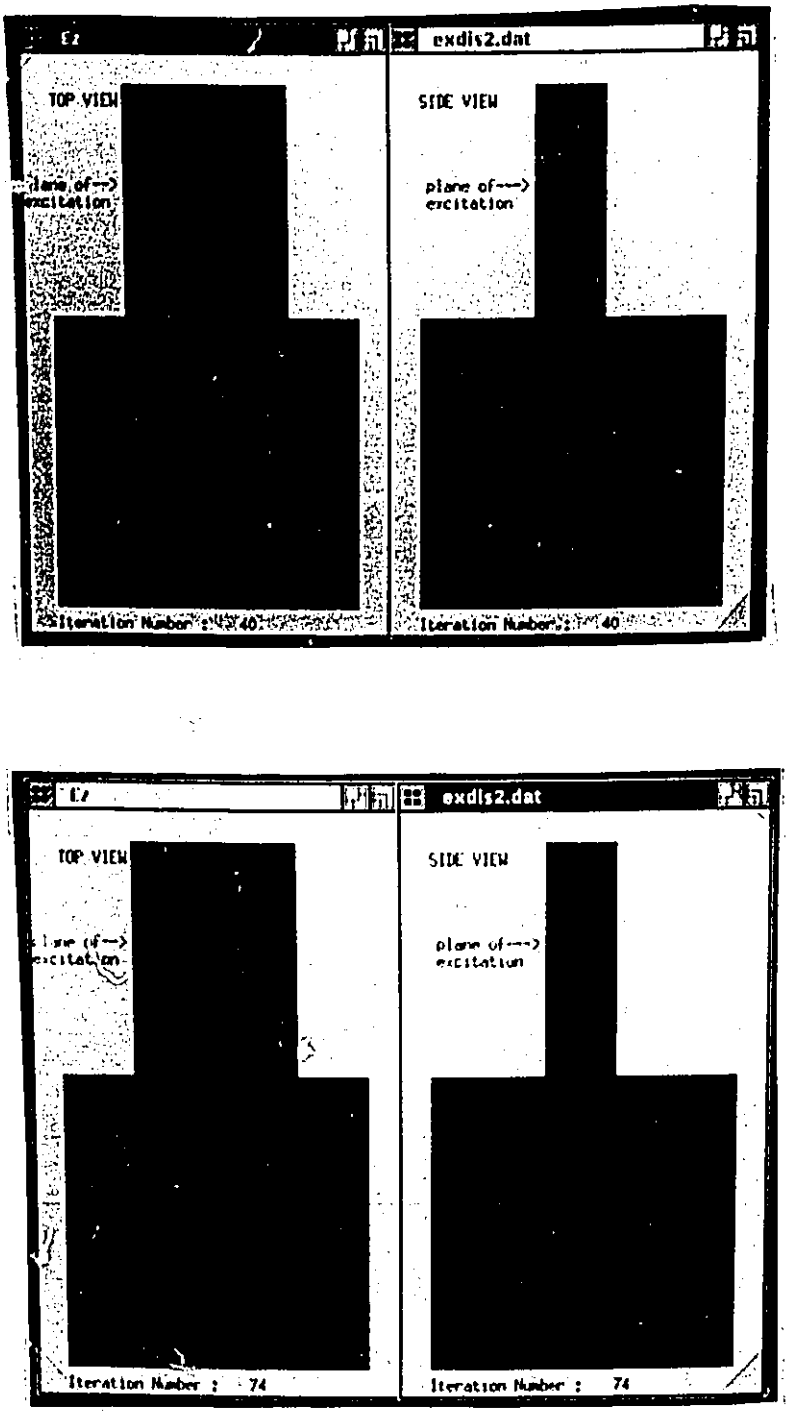


Figure 4.5: Top and side view photographs of E_z for iterations 40 and 74

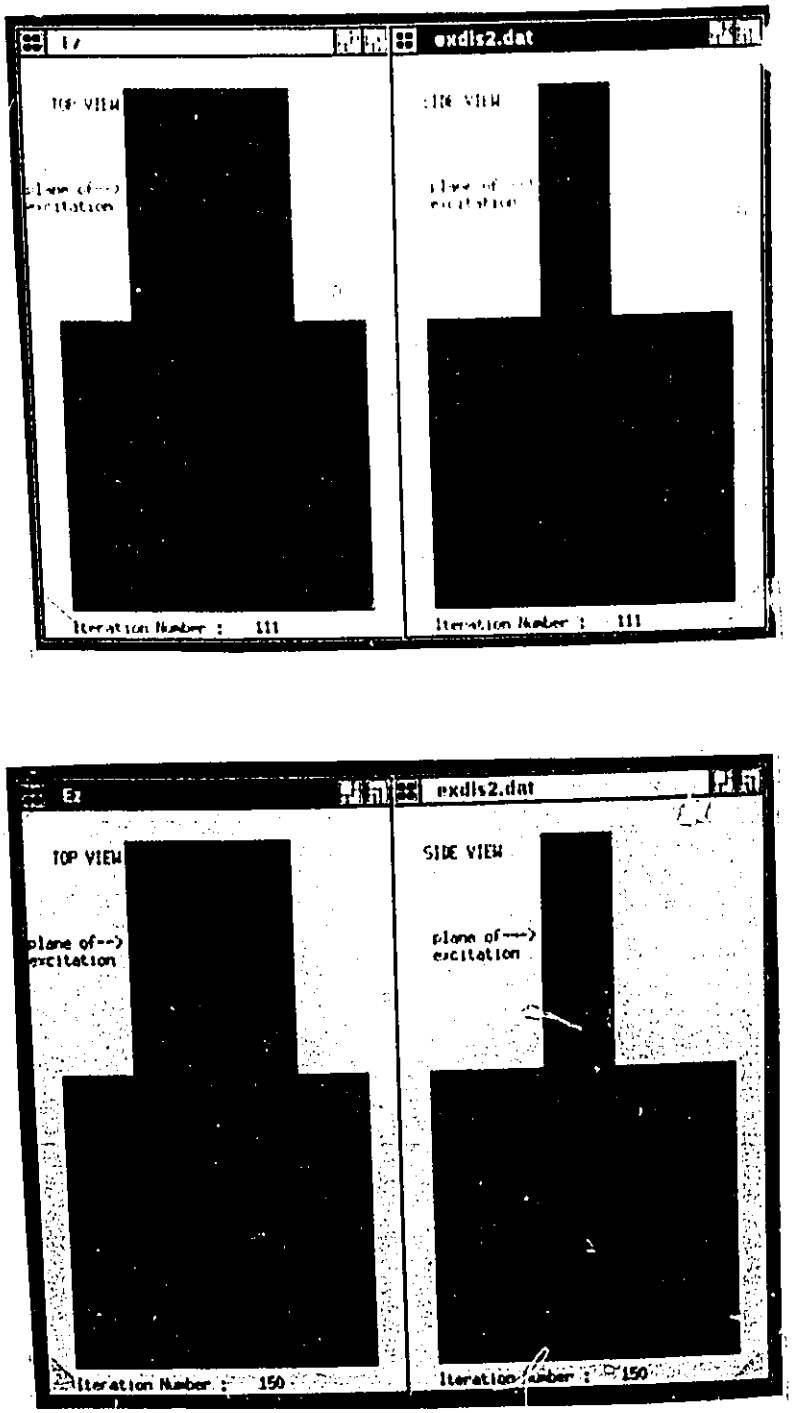


Figure 4.6: Top and side view photographs of E_z for iterations 111 and 150

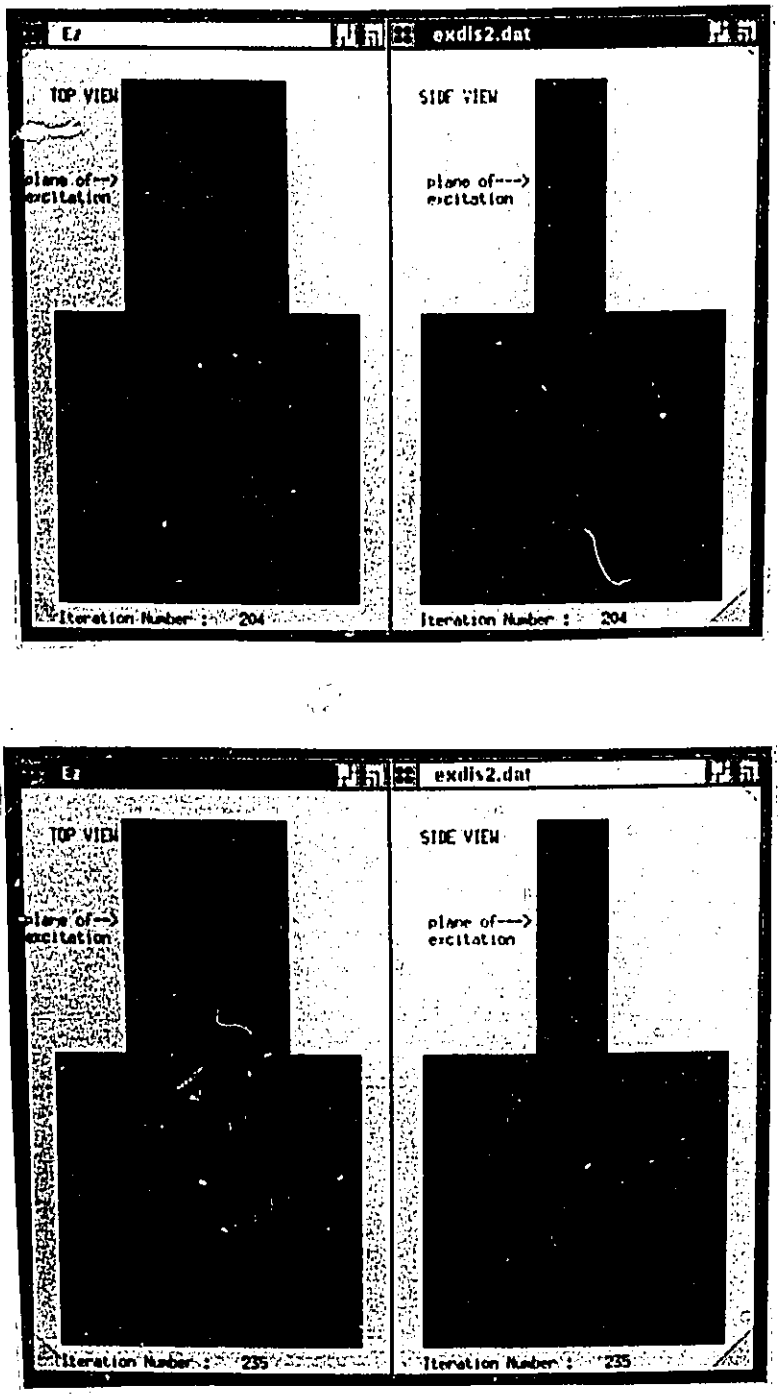


Figure 4.7: Top and side view photographs of E_z for iterations 204 and 235

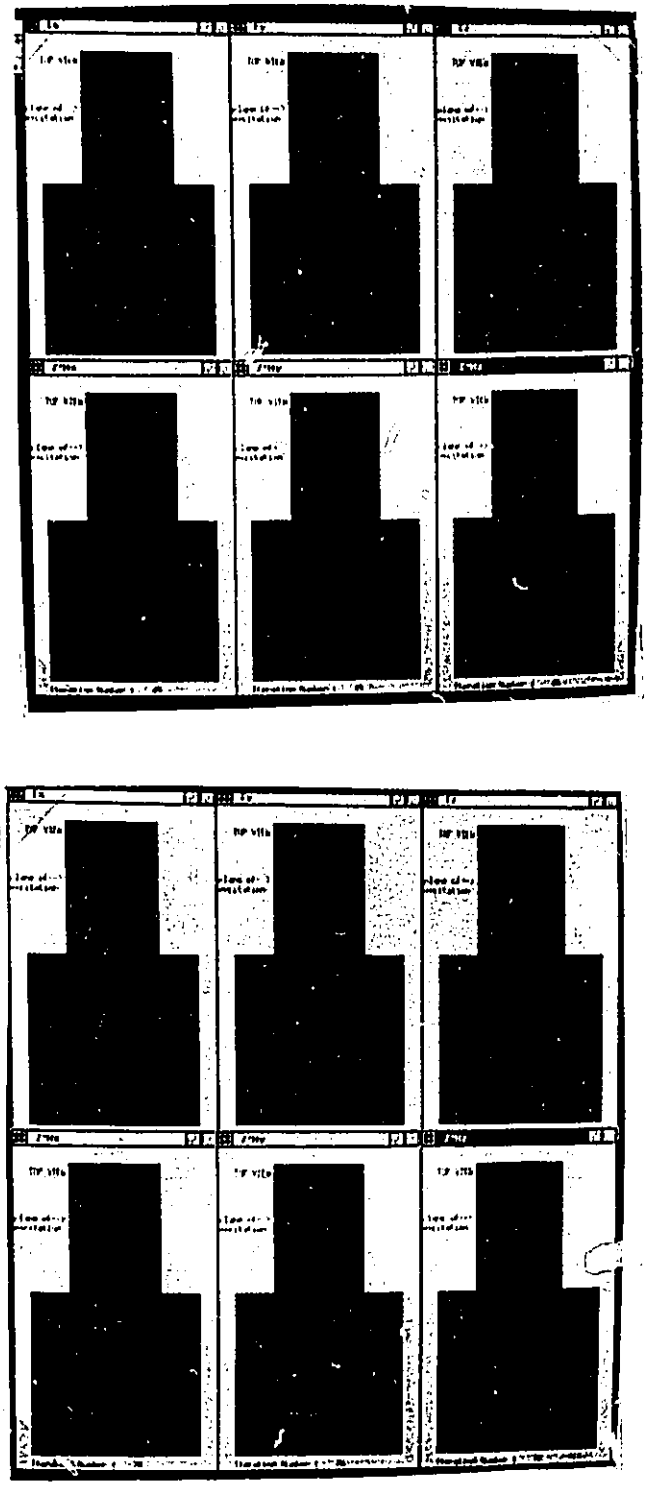


Figure 4.8: Top view photographs of $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ for iterations 45 and 76

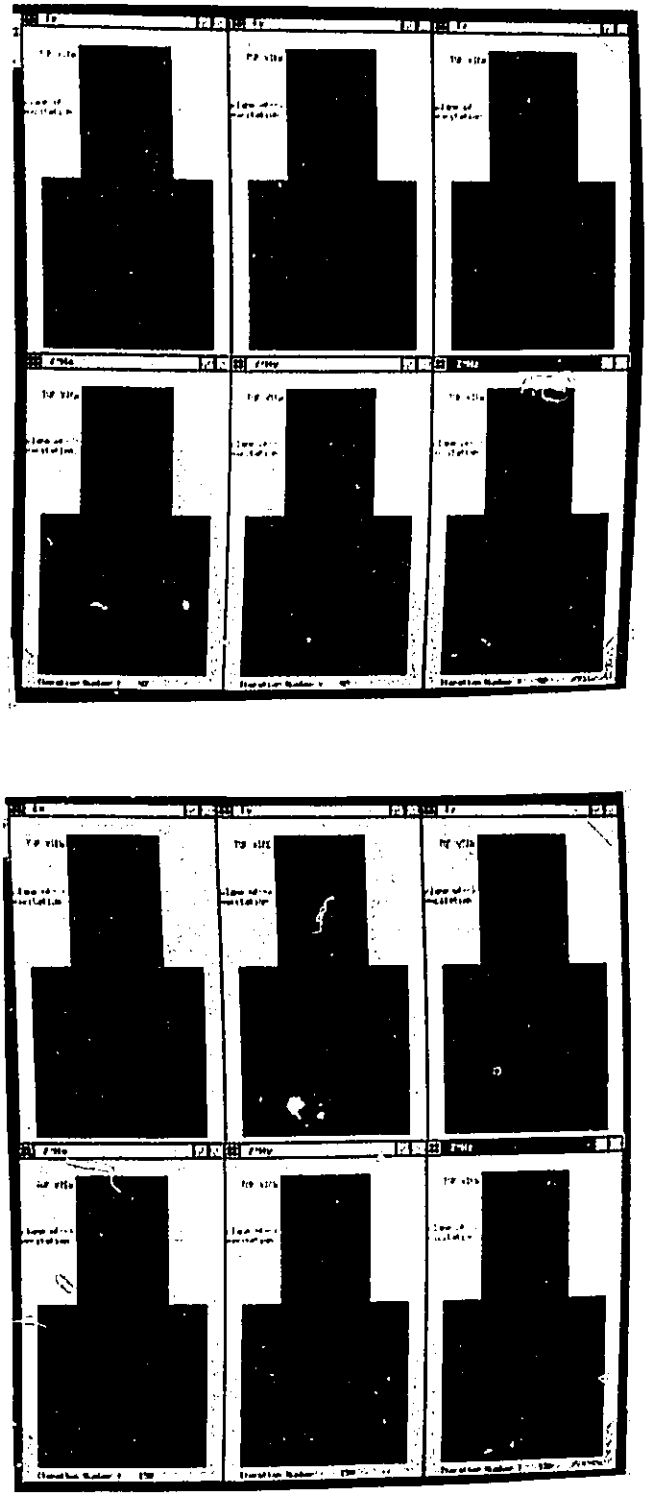


Figure 4.9: Top view photographs of $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ for iterations 92 and 120

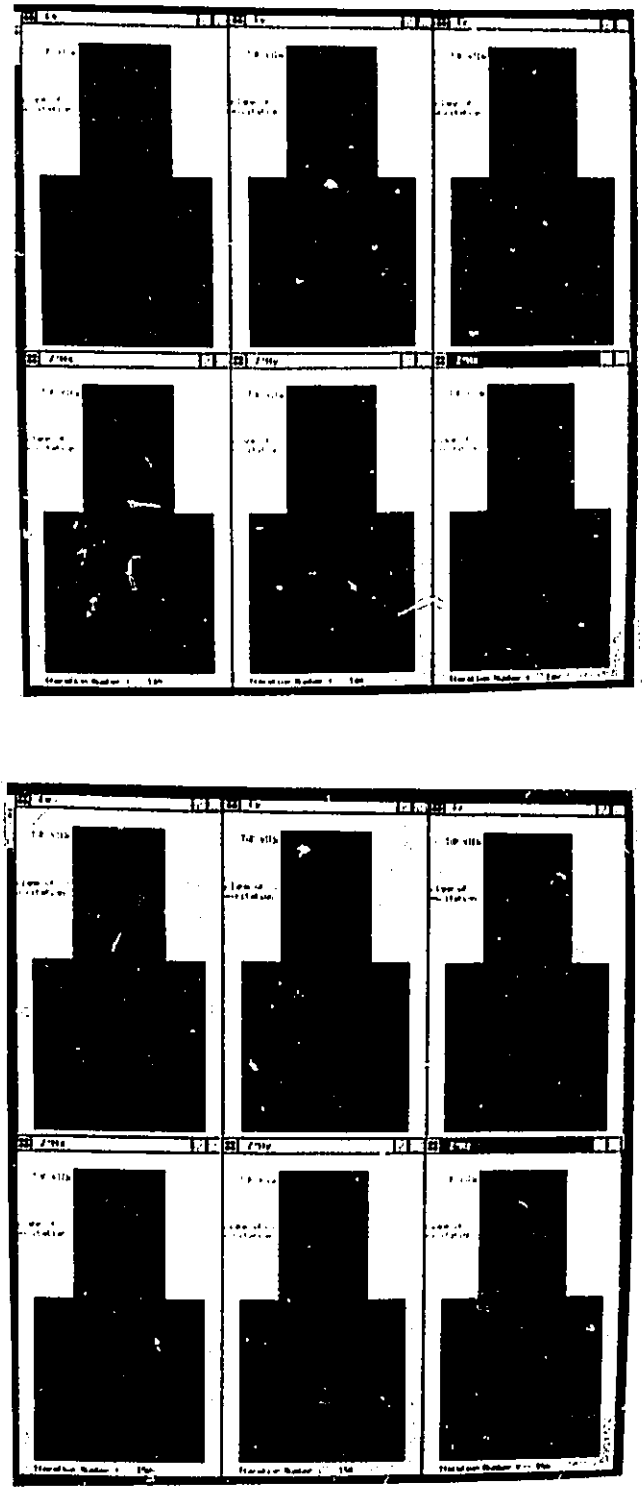


Figure 4.10: Top view photographs of $\vec{\mathbf{E}}$ and $\vec{\mathbf{H}}$ for iterations 140 and 150

Chapter 5

Conclusions

The original intention of the author was to validate the FDTD method by comparing the Γ obtained from measurements and simulation of a realistic microwave structure. For reasons discussed in chapter 3.4, the agreement was not exact but, in the author's opinion, enough to validate the method.

In retrospect, had the region in the simulation over which the standing-wave pattern was determined been longer, perhaps the variations in the pattern would have shown themselves to be cyclical or bounded in some way. This might have allowed one to average the values of the maximums and minimums and obtain more accurate values for the $VSWR$ and Γ. Unfortunately, due to computer limitations, it was not practical to increase the size of the structure being simulated; in a sense, it was already too large for the available computers.

It could well be argued that the method for calculating Γ was not well suited to the nature of the simulation and that, again in retrospect, another approach should have been used. Regardless of the difficulties encountered in determining Γ from simulation data, the author is convinced that adequate agreement exists between measured and simulated data to conclude that the method is sound and its results reliable. This finding is further supported by the symmetrical behavior of the EM fields as can be seen in the photos of chapter 4.2.

The single greatest hinderance and ultimate limitation that the author repeatedly encountered throughout this study was the computer itself. Numerous authors have commented upon the numerical burden placed upon computers by Yee's FDTD scheme. This method is slightly more onerous for computers, as was first mentioned in chapter 1.2, since each unit cell has 30 fields components that must be calculated for each iteration (6 node fields and 24 half fields - see Figure 2.1). The computer

that was used most often in this study for simulation was the IBM/RS 6000. This machine, with no other users (which was dishearteningly rare), required about 16 hours to complete an EXPAR simulation. The author tried to run the simulation on other machines in the hope of reducing the run-time but to no avail. The University of Ottawa's mainframe computer (at the time an AMDAHL 5880) and the CRAY-XMP computer at the University of Toronto's Center for Large Scale Computing were both able to compile the program which contained the simulation but unable to run it reporting insufficient memory.

In [2], the author discusses the improvement upon performance that can be obtained when the computer algorithm is optimized for a given computer. This is an important aspect of any computer simulation since a well written computer program which efficiently utilizes the resources of the machine can run faster than a program written with no consideration given to the computer's hardware.

The computer program (or source code) EXPAR3, which is the simulation of the EXPAR3 structure, is included in this study as Appendix B (the program used to generate the standing-wave pattern is included as Appendix C). The problem space was discretized in a manner that would enhance the run-time: it was found that most computers could more quickly process many smaller arrays than fewer larger ones. This was the extent of the optimization that was performed because the author was attempting to run the simulation on various machines and therefore could not focus on a single computer for optimization. It was not apparent until late in the course of this study that, of the available computers, only the IBM/RS 6000 would be capable of running the simulation. Furthermore, it was felt that little could be accomplished by a lengthy and detailed study of IBM/RS 6000 architecture; the structure was just so large to begin with that only by making it smaller could the run-time be significantly reduced. In the EXPAR3 structure there existed 110,484 nodes. Approximately 663,000 vector field components (node fields) and some 2.65M half fields were recalculated for each iteration.

It is the author's opinion that with the use of colour monitors such as the DEC station 3100, this method readily lends itself to displaying radiation characteristics of microwave devices and antennas. Indeed, the iridescent capabilities of modern computers has greatly enhanced the presentability of EM field data. This approach has already been used to a certain degree in [31]. The author would advocate using field data provided by the simulation to calculate power flow (i.e. Poynting vector)

accompanied by a colour scale. This type of display could show a user where the power in a microwave structure or antenna is going and give an estimate on relative power levels throughout the structure.

By way of further study, the author recommends that the variation in the standing-wave pattern be investigated in greater detail. It may well be that determining the standing-wave pattern via a numerical simulation is not appropriate and will always be subject to this type of error.

Bibliography

[1] Yee, K.S., "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media",*IEEE Transactions on Antennas and Propagation*, Vol. AP-14, pp. 302-307, May 1966.

[2] Taflove, A.,"State of the Art and Future Directions in Finite-Difference and Related Techniques in Supercomputing Computational Electromagnetics", Ed. H.L. Bertoni and L.B. Felsen, *Directions in Electromagnetic Wave modeling*, Plenum Press, New York, 1991.

[3] Taflove, A., and K.R. Umashankar, "Review of FD-TD Numerical Modeling of Electromagnetic Wave Scattering and Radar Cross Section",*Proceedings of the IEEE*, no. 5, May 1989.

[4] Kim, I.S., *Contributions to the Time Domain-Finite Difference Method for the Modeling of Microwave Structures*, Ph.D. Dissertation, University of Ottawa, 1990.

[5] Holland, R., "THREDE: A Free-Field EMP Coupling and Scattering Code", *IEEE Transactions on Nuclear Science*, Vol. NS-24, no. 6, pp.2416-2421, December 1977.

[6] Kunz, K.S., and Kuan-Min Lee,"A Three-Dimensional Finite-Difference Solution of the External Response of an Aircraft to a Complex Transient EM Environment: Part I - The Method and Its Implementation", *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-20, no. 2, pp. 328-333, May 1978.

[7] Holland, R., et al., "Finite-Difference Analysis of EMP Coupling to Lossy Dielectric Structures", *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-22, no. 3, pp.203-209, August 1980.

[8] Holland, R., and L. Simpson, "Finite-Difference Analysis of EMP Coupling to Thin Struts and Wires", *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-23, n. 2, pp. 88-97, May 1981.

[9] Chen, Jin-Yuan, and O.P. Gandhi, "Currents Induced in an Anatomically Based Model of a Human for Exposure to Vertically Polarized Electromagnetic Pulses", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 39, no. 1, pp. 31-39, January 1991.

[10] Sullivan, D., "Three-Dimensional Computer Simulation in Deep Regional Hyperthermia Using the Finite-Difference Time-Domain Method", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 38, no. 2, pp. 204-211, February 1990.

[11] Sullivan, D., et al., "Use of the Finite-Difference Time-Domain Method in Calculating EM Absorption in Human Tissues", *IEEE Transactions on Biomedical Engineering*, Vol. BME-34, no. 2, pp. 148-157, February 1987.

[12] Wang, Chang-Qing, and O.P. Gandhi, "Numerical Simulation of Annular Phased Arrays for Anatomically Based Models Using the FDTD Method", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 37, no. 1, pp.118-126, January 1989.

[13] Moore, J., and H. Ling, "Characterization of a 90° Microstrip Bend with Arbitrary Miter via the Time-Domain Finite Difference Method", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 38, no. 4, pp. 405-409, April 1990.

[14] Wolff, I., and M. Rittweger, "Finite Difference Time-Domain Analysis of Planar Microwave Circuits", *Archiv fur Elektrotechnik*, Vol. 74, pp. 189-201, 1991.

[15] Zhang, X., and K.K. Mei, "Time-Domain Finite Difference Approach to the Calculation of the Frequency-Dependent Characteristics of Microstrip Discontinuities", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 36, no. 12, pp. 1775-1787, December 1988.

[16] Umashankar, K., and A. Taflove, "A Novel Method to Analyze Electromagnetic Scattering of Complex Objects", *IEEE Transactions on Electromagnetic Compatibility*, Vol. EMC-24, no. 4, pp. 397-405, November 1982.

[17] Chen, Z., et al., "A New Finite-Difference Time-Domain Formulation and its Equivalence with the TLM Symmetrical Condensed Node", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 39, no. 12, pp. 2160-2169, December 1991.

[18] Choi, D.K., and W. Hoefer, "The Finite-Difference Time-Domain Method and its Application to Eigenvalue Problems", *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-34, no. 12, pp. 1464-1469, December 1986.

[19] Taflove, A., and M.E. Brodwin, "Numerical Solution of Steady-State Electromagnetic Scattering Problems Using the Time-Dependent Maxwell's Equations", *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-23, no. 8, pp.623-630, August 1975.

[20] Haberman, R., *Elementary Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*, Prentice-Hall, New Jersey, 1983.

[21] Chen, Z., *The Transmission Line Matrix Method and its Boundary Treatment*, Ph.D. Dissertation, University of Ottawa, July 1992.

[22] Saguet, P., "TLM Method for the Three Dimensional Analysis of Microwave and mm-Wave Structures", *International Workshop of the German IEEE MTT/AP Chapter*, Stuttgart, September 13, 1991.

[23] Nielsen, J., "Spurious Modes of the TLM-Condensed Node Formulation", *IEEE Microwave and Guided Wave Letters*, Vol. 1, no. 8, pp.201-203, August 1991.

[24] Liao, S.Y., *Microwave Circuit Analysis and Amplifier Design*, Prentice-Hall, New Jersey, 1987.

[25] Gardiol, F.E., "Open-Ended Waveguides: Principles and Applications", Ed. P.W Hawkes, *Advances in Electronics and Electron Physics*, Vol. 63, Academic Press, 1985.

[26] Gardiol, F.E., *Introduction to Microwaves*, Artech House,Dedham, MA, 1984.

[27] Harrington, R.F., *Time-Harmonic Electromagnetic Fields*, McGraw-Hill, Toronto, 1961.

[28] Nielsen, J., and W. Hoefer, "A Complete Dispersion Analysis of the Condensed Node TLM Mesh", *4th Biennial IEEE Conference on EM Field Computaion*, Tornonto, Ont., Oct 1990.

[29] King, R.W.P., and S. Prasad, *Fundamental Electromagnetic Theory and Applications*, Prentice-Hall, Englewood Cliffs, N.J., 1986.

[30] Zhang, X., et al., "Calculations of the Dispersive Characteristics of Microstrips by Time-Domain Finite Difference Method",*IEEE Transactions on Microwave Theory and Techniqaes*, Vol. 36, pp. 263- 267, February 1988.

[31] Maloney, J., et al., "Accurate Computation of the Radiation From Simple Antennas Using the Finite-Difference Time-Domain Method", *1989 IEEE AP-S International Symposium (Digest)*, Vol. 1, pp. 42-45, San Jose, CA, June 26-30, 1989.

Appendix A

FDTD Equations for the Absorbing Wall (AW)

This appendix contains the boundary equations for the absorbing walls that were used in this study (i.e. the AW boundary equations). These equations, combined with the appropriate boundary equations (equations 2.21 to 2.32), can be easily manipulated to form explicit expressions for E and H half fields that are tangential to the AW boundary surface (i.e. AW half fields) as was discused in chapter 2.3.

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o = \\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_x(i,j-\frac{3}{2},k) + {}_{n-\frac{7}{2}}H_z(i,j-\frac{3}{2},k)Z_o\right] \\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_x(i,j-\frac{7}{2},k) + {}_{n-\frac{15}{2}}H_z(i,j-\frac{7}{2},k)Z_o\right] \\
&\quad 0.25\left[{}_{n-\frac{23}{2}}E_x(i,j-\frac{11}{2},k) + {}_{n-\frac{23}{2}}H_z(i,j-\frac{1}{2},k)Z_o\right]
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) - {}_{n+\frac{1}{2}}H_x(i,j+\frac{1}{2},k)Z_o = \\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_z(i,j-\frac{3}{2},k) - {}_{n-\frac{7}{2}}H_x(i,j-\frac{3}{2},k)Z_o\right] \\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_z(i,j-\frac{7}{2},k) - {}_{n-\frac{15}{2}}H_x(i,j-\frac{7}{2},k)Z_o\right] \\
&\quad 0.25\left[{}_{n-\frac{23}{2}}E_z(i,j-\frac{11}{2},k) - {}_{n-\frac{23}{2}}H_x(i,j-\frac{11}{2},k)Z_o\right]
\end{aligned}
\tag{A.2}
$$

$$
{}_{n+\frac{1}{2}}E_z(i,j+\frac{1}{2},k) + {}_{n+\frac{1}{2}}H_x(i,j+\frac{1}{2},k)Z_o = \tag{A.3}
$$

$$2.25\left[{}_{n-\frac{7}{2}}E_z(i,j+\frac{5}{2},k)+{}_{n-\frac{7}{2}}H_x(i,j+\frac{5}{2},k)Z_o\right]$$
$$-1.5\left[{}_{n-\frac{15}{2}}E_z(i,j+\frac{9}{2},k)+{}_{n-\frac{15}{2}}H_x(i,j+\frac{9}{2},k)Z_o\right]$$
$$0.25\left[{}_{n-\frac{23}{2}}E_z(i,j+\frac{13}{2},k)+{}_{n-\frac{23}{2}}H_x(i,j+\frac{13}{2},k)Z_o\right]$$

$${}_{n+\frac{1}{2}}E_x(i,j+\frac{1}{2},k)-{}_{n+\frac{1}{2}}H_z(i,j+\frac{1}{2},k)Z_o= \tag{A.4}$$
$$2.25\left[{}_{n-\frac{7}{2}}E_x(i,j+\frac{5}{2},k)-{}_{n-\frac{7}{2}}H_z(i,j+\frac{5}{2},k)Z_o\right]$$
$$-1.5\left[{}_{n-\frac{15}{2}}E_x(i,j+\frac{9}{2},k)-{}_{n-\frac{15}{2}}H_z(i,j+\frac{9}{2},k)Z_o\right]$$
$$0.25\left[{}_{n-\frac{23}{2}}E_x(i,j+\frac{13}{2},k)-{}_{n-\frac{23}{2}}H_z(i,j+\frac{13}{2},k)Z_o\right]$$

$${}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2})+{}_{n+\frac{1}{2}}H_x(i,j,k+\frac{1}{2})Z_o= \tag{A.5}$$
$$2.25\left[{}_{n-\frac{7}{2}}E_y(i,j,k-\frac{3}{2})+{}_{n-\frac{7}{2}}H_x(i,j,k-\frac{3}{2})Z_o\right]$$
$$-1.5\left[{}_{n-\frac{15}{2}}E_y(i,j,k-\frac{7}{2})+{}_{n-\frac{15}{2}}H_x(i,j,k-\frac{7}{2})Z_o\right]$$
$$0.25\left[{}_{n-\frac{23}{2}}E_y(i,j,k-\frac{11}{2})+{}_{n-\frac{23}{2}}H_x(i,j,k-\frac{11}{2})Z_o\right]$$

$${}_{n+\frac{1}{2}}E_x(i,j,k+\frac{1}{2})-{}_{n+\frac{1}{2}}H_y(i,j,k+\frac{1}{2})Z_o= \tag{A.6}$$
$$2.25\left[{}_{n-\frac{7}{2}}E_x(i,j,k-\frac{3}{2})-{}_{n-\frac{7}{2}}H_y(i,j,k-\frac{3}{2})Z_o\right]$$
$$-1.5\left[{}_{n-\frac{15}{2}}E_x(i,j,k-\frac{7}{2})-{}_{n-\frac{15}{2}}H_y(i,j,k-\frac{7}{2})Z_o\right]$$
$$0.25\left[{}_{n-\frac{23}{2}}E_x(i,j,k-\frac{11}{2})-{}_{n-\frac{23}{2}}H_y(i,j,k-\frac{11}{2})Z_o\right]$$

$${}_{n+\frac{1}{2}}E_y(i,j,k+\frac{1}{2})-{}_{n+\frac{1}{2}}H_x(i,j,k+\frac{1}{2})Z_o= \tag{A.7}$$
$$2.25\left[{}_{n-\frac{7}{2}}E_y(i,j,k+\frac{5}{2})-{}_{n-\frac{7}{2}}H_x(i,j,k+\frac{5}{2})Z_o\right]$$
$$-1.5\left[{}_{n-\frac{15}{2}}E_y(i,j,k+\frac{9}{2})-{}_{n-\frac{15}{2}}H_x(i,j,k+\frac{9}{2})Z_o\right]$$
$$0.25\left[{}_{n-\frac{23}{2}}E_y(i,j,k+\frac{13}{2})-{}_{n-\frac{23}{2}}H_x(i,j,k+\frac{13}{2})Z_o\right]$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_x(i,j,k+\frac{1}{2}) +_{n+\frac{1}{2}} H_y(i,j,k+\frac{1}{2})Z_o = \qquad (A.8)\\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_x(i,j,k+\frac{5}{2}) +_{n-\frac{7}{2}} H_y(i,j,k+\frac{5}{2})Z_o\right]\\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_x(i,j,k+\frac{9}{2}) +_{n-\frac{15}{2}} H_y(i,j,k+\frac{9}{2})Z_o\right]\\
&\quad 0.25\left[{}_{n-\frac{23}{2}}E_x(i,j,k+\frac{13}{2}) +_{n-\frac{23}{2}} H_y(i,j+\frac{13}{2},k)Z_o\right]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_z(i+\frac{1}{2},j,k) +_{n+\frac{1}{2}} H_y(i+\frac{1}{2},j,k)Z_o = \qquad (A.9)\\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_z(i-\frac{3}{2},j,k) +_{n-\frac{7}{2}} H_y(i-\frac{3}{2},j,k)Z_o\right]\\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_z(i-\frac{7}{2},j,k) +_{n-\frac{15}{2}} H_y(i-\frac{7}{2},j,k)Z_o\right]\\
&\quad 0.25\left[{}_{n-\frac{23}{2}}E_z(i-\frac{11}{2},j,k) +_{n-\frac{23}{2}} H_y(i-\frac{11}{2},j,k)Z_o\right]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_y(i+\frac{1}{2},j,k) -_{n+\frac{1}{2}} H_z(i+\frac{1}{2},j,k)Z_o = \qquad (A.10)\\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_y(i-\frac{3}{2},j,k) -_{n-\frac{7}{2}} H_z(i-\frac{3}{2},j,k)Z_o\right]\\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_y(i-\frac{7}{2},j,k) -_{n-\frac{15}{2}} H_z(i-\frac{7}{2},j,k)Z_o\right]\\
&\quad 0.25\left[{}_{n-\frac{23}{2}}E_y(i-\frac{11}{2},j,k) -_{n-\frac{23}{2}} H_z(i-\frac{11}{2},j,k)Z_o\right]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_z(i+\frac{1}{2},j,k) -_{n+\frac{1}{2}} H_y(i+\frac{1}{2},j,k)Z_o = \qquad (A.11)\\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_z(i+\frac{5}{2},j,k) -_{n-\frac{7}{2}} H_y(i+\frac{5}{2},j,k)Z_o\right]\\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_z(i+\frac{9}{2},j,k) -_{n-\frac{15}{2}} H_y(i+\frac{9}{2},j,k)Z_o\right]\\
&\quad 0.25\left[{}_{n-\frac{23}{2}}E_z(i+\frac{13}{2},j,k) -_{n-\frac{23}{2}} H_y(i+\frac{13}{2},j,k)Z_o\right]
\end{aligned}
$$

$$
\begin{aligned}
&{}_{n+\frac{1}{2}}E_y(i+\frac{1}{2},j,k) +_{n+\frac{1}{2}} H_z(i+\frac{1}{2},j,k)Z_o = \qquad (A.12)\\
&\quad 2.25\left[{}_{n-\frac{7}{2}}E_y(i+\frac{5}{2},j,k) +_{n-\frac{7}{2}} H_z(i+\frac{5}{2},j,k)Z_o\right]\\
&\quad -1.5\left[{}_{n-\frac{15}{2}}E_y(i+\frac{9}{2},j,k) +_{n-\frac{15}{2}} H_z(i+\frac{9}{2},j,k)Z_o\right]
\end{aligned}
$$

$$0.25\left[{}_{n-\frac{23}{2}}E_y(i+\frac{13}{2},j,k)+{}_{n-\frac{23}{2}}H_z(i+\frac{13}{2},j,k)Z_0\right]$$

Appendix B

EXPAR3 Computer Simulation

This program written in C represents the FDTD simulation of the EXPAR structure. The entire problem space is divided up into three regions: waveguide, dielectric and free space. In each of these regions, arrays exists to hold node and half field values as well as special arrays required for the operation of the absorbing walls and excitation functions. Each node and half field array has four dimensions: time and the three space indices (i, j, k). In order to save memory, only the time data necessary for the operation of the algorithm is kept. Field values for a future time step supercede values for the present time step once the simulation advances. In this manner, new fields values are exchanged for previous ones and old values are discarded. The output files, exdis1.dat and exdis2.dat, are used to store data for the colour display program. They have been commented out in this listing. (A comment in C begins and ends with the character sequences /* and */ repectively. The output file exparfft.dat contains the E_z contibution to the TE_{10} mode in the cross section of the guide from j=20 to j=70 for each iterative step.

```
#include <stdio.h>
#include <math.h>
FILE *output1,*output2,*output3,*fopen();
main()
{
float  a,aa,er,ur,erd,urd,dumm1,dumm2;
int t,i,j,k,imax,jmax,kmax,tmax,n,step,imaxd,jmaxd,kmaxd;
int imaxf,jmaxf,kmaxf,NSTEP;

/* Arrays for the waveguide section  */
static float  Ex[4][25][122][12];
static float  Ey[4][25][122][12];
```

```
static float  Ez[4][25][122][12];
static float  Bx[4][25][122][12];
static float  By[4][25][122][12];
static float  Bz[4][25][122][12];
static float  njEx[4][25][122][12];
static float  nkEx[4][25][122][12];
static float  niEy[4][25][122][12];
static float  nkEy[4][25][122][12];
static float  niEz[4][25][122][12];
static float  njEz[4][25][122][12];
static float  njBx[4][25][122][12];
static float  nkBx[4][25][122][12];
static float  niBy[4][25][122][12];
static float  nkBy[4][25][122][12];
static float  niBz[4][25][122][12];
static float  njBz[4][25][122][12];

static float  njEzexcit[25][12];   /* These two arrays are for the Gaussian pulse */
static float  njBxexcit[25][12];   /* excitation                                  */

static float  njE1x[11][25][3][12]; /* These next 4 arrays are used to store*/
static float  njE1z[11][25][3][12]; /* values for the absorbing wall in the */
static float njB1x[11][25][3][12];  /* waveguide region.                    */
static float  njB1z[11][25][3][12];

/* Arrays for the dielectric region  */
static float  Exd[4][25][4][12];
static float  Eyd[4][25][4][12];
static float  Ezd[4][25][4][12];
static float  Bxd[4][25][4][12];
static float  Byd[4][25][4][12];
static float  Bzd[4][25][4][12];

static float  njExd[4][25][4][12];
static float  nkExd[4][25][4][12];
static float  niEyd[4][25][4][12];
static float  nkEyd[4][25][4][12];
static float  niEzd[4][25][4][12];
static float  njEzd[4][25][4][12];
static float  njBxd[4][25][4][12];
static float  nkBxd[4][25][4][12];
static float  niByd[4][25][4][12];
static float  nkByd[4][25][4][12];
static float  niBzd[4][25][4][12];
static float  njBzd[4][25][4][12];

/* Arrays for the free space region  */
static float  Exf[4][45][45][45];
static float  Eyf[4][45][45][45];
static float  Ezf[4][45][45][45];
static float  Bxf[4][45][45][45];
```

```
static float  Byf[4][45][45][45];
static float  Bzf[4][45][45][45];

static float  njExf[4][45][45][45];
static float  nkExf[4][45][45][45];
static float  niEyf[4][45][45][45];
static float  nkEyf[4][45][45][45];
static float  niEzf[4][45][45][45];
static float  njEzf[4][45][45][45];
static float  njBxf[4][45][45][45];
static float  nkBxf[4][45][45][45];
static float  niByf[4][45][45][45];
static float  nkByf[4][45][45][45];
static float  niBzf[4][45][45][45];
static float  njBzf[4][45][45][45];
static float  njE1xf[11][45][3][45];  /* These next 8 arrays are   */
static float  njE1zf[11][45][3][45];  /* used to store values for  */
static float  njB1xf[11][45][3][45];  /* the absorbing walls along j*/
static float  njB1zf[11][45][3][45];
static float  njE2xf[11][45][3][45];
static float  njE2zf[11][45][3][45];
static float  njB2xf[11][45][3][45];
static float  njB2zf[11][45][3][45];

static float  niE1yf[11][3][45][45];  /* These next 8 arrays are   */
static float  niE1zf[11][3][45][45];  /* used to store values for  */
static float  niB1yf[11][3][45][45];  /* the absorbing walls along i*/
static float  niB1zf[11][3][45][45];
static float  niE2yf[11][3][45][45];
static float  niE2zf[11][3][45][45];
static float  niB2yf[11][3][45][45];
static float  niB2zf[11][3][45][45];

static float  nkE1yf[11][45][45][3];  /* These next 8 arrays are   */
static float  nkE1xf[11][45][45][3];  /* used to store values for  */
static float  nkB1yf[11][45][45][3];  /* the absorbing walls along k*/
static float  nkB1xf[11][45][45][3];
static float  nkE2yf[11][45][45][3];
static float  nkE2xf[11][45][45][3];
static float  nkB2yf[11][45][45][3];
static float  nkB2xf[11][45][45][3];

NSTEP=1000;/* NSTEP is the number of iterations in time.  */
er=2.0;    /* "Free Space" values for the purposes of this simulation  */
ur=2.0;
erd=4.66;   /* These are the values for the dielectric region. In reality they */
urd=2.0;    /* are ur=1 and er=2.33.  Output data must be scaled appropriately.*/
/***********************************************/
/*  Note: tmax,imax,jmax and kmax should all be */
/*  one less than t,i,j,k as shown in the above */
/*  declaration statements for Ex,Ey,Ez,Bx,By   */
```

```
/*  and Bz.                                          */
/*************************************************/
tmax=3;
imax=24;   /* these values for the waveguide region  */
jmax=121;
kmax=11;
imaxd=24;   /* these values for the dielectric region  */
jmaxd=3;
kmaxd=11;
imaxf=44;   /* these values for the free space region  */
jmaxf=44;
kmaxf=44;
output1=fopen("exdis1.dat","w");
output2=fopen("exdis2.dat","w");
output3=fopen("exparfft.dat","w");
printf("FD-TD program EXPAR3 has begun.\n\n");
n=12; /* time step for absorbing arrays  */
t=1;  /* this is a time index for the simulation */
/*****************************************************************/
/*    Start tangential field calculations for the next half time */
/*    step (ie at t+1) for all nodes in the entire structure     */
/*****************************************************************/
for (step=1;step<=NSTEP;++step)
{
    printf("%d of %d\n",step,NSTEP);
/*****************************************************************/
/* This section of code controls the excitation of the guide   */
/* which occurs in the x-z plane at j=15.  The  Ez and Bx field*/
/* components are excited in a sinusiodal and Gaussian manner  */
/* in space and time respectively.  This results in TE10 mode  */
/* propagation.                                                */
/*****************************************************************/
for (k=1;k<=kmax-1;++k)
    {   for (i=1;i<=imax-1;++i)
        {   njEzexcit[i][k]=exp(-((step-40.0)/10.0)*((step-40.0)/10.0))
                            *sin(M_PI*(i-0.5)/(imax-1));
            njBxexcit[i][k]=exp(-((step-40.0)/10.0)*((step-40.0)/10.0))
                             *sin(M_PI*(i-0.5)/(imax-1));
        }
    }
/*****************************************************************/
/* First calculate all half fields tangentail to the boundaries*/
/* of the waveguide region.                                    */
/*****************************************************************/
/* E wall boundary on the bottom  */
    k=0;
    for (j=1;j<=(jmax-1);++j)
    {   for (i=1;i<=(imax-1);++i)
        {   nkEx[t+1][i][j][k]=0.0;
            nkBy[t+1][i][j][k]=(-2.0)*(Ex[t][i][j][k+1]-By[t][i][j][k+1])
                               +(nkEx[t-1][i][j][k+1]-nkBy[t-1][i][j][k+1]);
```

```
        nkEy[t+1][i][j][k]=0.0;
        nkBx[t+1][i][j][k]=(2.0)*(Ey[t][i][j][k+1]+Bx[t][i][j][k+1])
                           -(nkEy[t-1][i][j][k+1]+nkBx[t-1][i][j][k+1]);
    }
}
/* E wall boundary on the top  */
k=kmax-1;
for (j=1;j<=(jmax-1);++j)
{   for (i=1;i<=(imax-1);++i)
    {   nkEx[t+1][i][j][k]=0.0;
        nkBy[t+1][i][j][k]=(2.0)*(Ex[t][i][j][k]+By[t][i][j][k])
                           -(nkEx[t-1][i][j][k-1]+nkBy[t-1][i][j][k-1]);
        nkEy[t+1][i][j][k]=0.0;
        nkBx[t+1][i][j][k]=(-2.0)*(Ey[t][i][j][k]-Bx[t][i][j][k])
                           +(nkEy[t-1][i][j][k-1]-nkBx[t-1][i][j][k-1]);
    }
}
/* E wall boundary for the back   */
i=0;
for (j=1;j<=(jmax-1);++j)
{   for (k=1;k<=(kmax-1);++k)
    {   niEy[t+1][i][j][k]=0.0;
        niBz[t+1][i][j][k]=(-2.0)*(Ey[t][i+1][j][k]-Bz[t][i+1][j][k])
                           +(niEy[t-1][i+1][j][k]-niBz[t-1][i+1][j][k]);
        niEz[t+1][i][j][k]=0.0;
        niBy[t+1][i][j][k]=(2.0)*(Ez[t][i+1][j][k]+By[t][i+1][j][k])
                           -(niEz[t-1][i+1][j][k]+niBy[t-1][i+1][j][k]);
    }
}
/* E wall boundary for the front  */
i=imax-1;
for (j=1;j<=(jmax-1);++j)
{   for (k=1;k<=(kmax-1);++k)
    {   niEy[t+1][i][j][k]=0.0;
        niBz[t+1][i][j][k]=(2.0)*(Ey[t][i][j][k]+Bz[t][i][j][k])
                           -(niEy[t-1][i-1][j][k]+niBz[t-1][i-1][j][k]);
        niEz[t+1][i][j][k]=0.0;
        niBy[t+1][i][j][k]=(-2.0)*(Ez[t][i][j][k]-By[t][i][j][k])
                           +(niEz[t-1][i-1][j][k]-niBy[t-1][i-1][j][k]);
    }
}
/* Absorbing wall boundary to the left */
j=0;
for (i=1;i<=(imax-1);++i)
{   for (k=1;k<=(kmax-1);++k)
    {   njEx[t+1][i][j][k]=Ex[t][i][j+1][k]+Bz[t][i][j+1][k]
                           -0.5*(njEx[t-1][i][j+1][k]+njBz[t-1][i][j+1][k])
                           +1.125*(njE1x[n-4][i][0][k]-njB1z[n-4][i][0][k])
                           -0.75*(njE1x[n-8][i][1][k]-njB1z[n-8][i][1][k])
                           +0.125*(njE1x[n-12][i][2][k]-njB1z[n-12][i][2][k]);
        njBz[t+1][i][j][k]=Ex[t][i][j+1][k]+Bz[t][i][j+1][k]
```

```
                        -0.5*(njEx[t-1][i][j+1][k]+njBz[t-1][i][j+1][k])
                        -1.125*(njE1x[n-4][i][0][k]-njB1z[n-4][i][0][k])
                        +0.75*(njE1x[n-8][i][1][k]-njB1z[n-8][i][1][k])
                        -0.125*(njE1x[n-12][i][2][k]-njB1z[n-12][i][2][k]);
        njEz[t+1][i][j][k]=Ez[t][i][j+1][k]-Bx[t][i][j+1][k]
                        -0.5*(njEz[t-1][i][j+1][k]-njBx[t-1][i][j+1][k])
                        +1.125*(njE1z[n-4][i][0][k]+njB1x[n-4][i][0][k])
                        -0.75*(njE1z[n-8][i][1][k]+njB1x[n-8][i][1][k])
                        +0.125*(njE1z[n-12][i][2][k]+njB1x[n-12][i][2][k]);
        njBx[t+1][i][j][k]=(-1.0)*Ez[t][i][j+1][k]+Bx[t][i][j+1][k]
                        +0.5*(njEz[t-1][i][j+1][k]-njBx[t-1][i][j+1][k])
                        +1.125*(njE1z[n-4][i][0][k]+njB1x[n-4][i][0][k])
                        -0.75*(njE1z[n-8][i][1][k]+njB1x[n-8][i][1][k])
                        +0.125*(njE1z[n-12][i][2][k]+njB1x[n-12][i][2][k]);
    }
}
/* E wall at the right end of waveguide section, wall has  */
/* a hole in it. This simulates a resonant iris.           */
j=jmax-1;
for (i=1;i<=2;++i)
{   for (k=1;k<=(kmax-1);++k)
    {   njEx[t+1][i][j][k]=0.0;
        njBz[t+1][i][j][k]=(-2.0)*(Ex[t][i][j][k]-Bz[t][i][j][k])
                        +(njEx[t-1][i][j-1][k]-njBz[t-1][i][j-1][k]);
        njEz[t+1][i][j][k]=0.0;
        njBx[t+1][i][j][k]=(2.0)*(Ez[t][i][j][k]+Bx[t][i][j][k])
                        -(njEz[t-1][i][j-1][k]+njBx[t-1][i][j-1][k]);
    }
}
for (i=imax-2;i<=(imax-1);++i)
{   for (k=1;k<=(kmax-1);++k)
    {   njEx[t+1][i][j][k]=0.0;
        njBz[t+1][i][j][k]=(-2.0)*(Ex[t][i][j][k]-Bz[t][i][j][k])
                        +(njEx[t-1][i][j-1][k]-njBz[t-1][i][j-1][k]);
        njEz[t+1][i][j][k]=0.0;
        njBx[t+1][i][j][k]=(2.0)*(Ez[t][i][j][k]+Bx[t][i][j][k])
                        -(njEz[t-1][i][j-1][k]+njBx[t-1][i][j-1][k]);
    }
}
for (i=3;i<=(imax-3);++i)
{   for (k=1;k<=2;++k)
    {   njEx[t+1][i][j][k]=0.0;
        njBz[t+1][i][j][k]=(-2.0)*(Ex[t][i][j][k]-Bz[t][i][j][k])
                        +(njEx[t-1][i][j-1][k]-njBz[t-1][i][j-1][k]);
        njEz[t+1][i][j][k]=0.0;
        njBx[t+1][i][j][k]=(2.0)*(Ez[t][i][j][k]+Bx[t][i][j][k])
                        -(njEz[t-1][i][j-1][k]+njBx[t-1][i][j-1][k]);
    }
}
for (i=3;i<=(imax-3);++i)
{   for (k=kmax-2;k<=kmax-1;++k)
```

```
        {   njEx[t+1][i][j][k]=0.0;
            njBz[t+1][i][j][k]=(-2.0)*(Ex[t][i][j][k]-Bz[t][i][j][k])
                               +(njEx[t-1][i][j-1][k]-njBz[t-1][i][j-1][k]);
            njEz[t+1][i][j][k]=0.0;
            njBx[t+1][i][j][k]=(2.0)*(Ez[t][i][j][k]+Bx[t][i][j][k])
                               -(njEz[t-1][i][j-1][k]+njBx[t-1][i][j-1][k]);
        }
    }
/************************************************/
/*  Now calculate half field components on the   */
/*  inside of the waveguide region.              */
/************************************************/
   for (k=1;k<=(kmax-1);++k)
   {   for (j=1;j<=(jmax-1);++j)
       {   for (i=1;i<=(imax-1);++i)
           {
/* half fields on cube face to the right of i,j,k   */
if (j==jmax-1) goto skip1;
njEx[t+1][i][j][k]=Ex[t][i][j][k]-Bz[t][i][j][k]+Ex[t][i][j+1][k]
                   +Bz[t][i][j+1][k]-(njEx[t-1][i][j+1][k]+njBz[t-1][i][j+1][k]
                   +njEx[t-1][i][j-1][k]-njBz[t-1][i][j-1][k])/2;
njBz[t+1][i][j][k]=(-Ex[t][i][j][k])+Bz[t][i][j][k]+Ex[t][i][j+1][k]
                   +Bz[t][i][j+1][k]-(njEx[t-1][i][j+1][k]+njBz[t-1][i][j+1][k]
                   -njEx[t-1][i][j-1][k]+njBz[t-1][i][j-1][k])/2;
njEz[t+1][i][j][k]=Ez[t][i][j][k]+Bx[t][i][j][k]+Ez[t][i][j+1][k]
                   -Bx[t][i][j+1][k]-(njEz[t-1][i][j+1][k]-njBx[t-1][i][j+1][k]
                   +njEz[t-1][i][j-1][k]+njBx[t-1][i][j-1][k])/2;
njBx[t+1][i][j][k]=Ez[t][i][j][k]+Bx[t][i][j][k]-Ez[t][i][j+1][k]
                   +Bx[t][i][j+1][k]-(-njEz[t-1][i][j+1][k]+njBx[t-1][i][j+1][k]
                   +njEz[t-1][i][j-1][k]+njBx[t-1][i][j-1][k])/2;
skip1:;
/*half fields on cube face in front of i,j,k    */
if (i==imax-1) goto skip2;
niEy[t+1][i][j][k]=Ey[t][i][j][k]+Bz[t][i][j][k]+Ey[t][i+1][j][k]
                   -Bz[t][i+1][j][k]-(niEy[t-1][i+1][j][k]-niBz[t-1][i+1][j][k]
                   +niEy[t-1][i-1][j][k]+niBz[t-1][i-1][j][k])/2;
niBz[t+1][i][j][k]=Ey[t][i][j][k]+Bz[t][i][j][k]-Ey[t][i+1][j][k]
                   +Bz[t][i+1][j][k]-(-niEy[t-1][i+1][j][k]+niBz[t-1][i+1][j][k]
                   +niEy[t-1][i-1][j][k]+niBz[t-1][i-1][j][k])/2;
niEz[t+1][i][j][k]=Ez[t][i][j][k]-By[t][i][j][k]+Ez[t][i+1][j][k]
                   +By[t][i+1][j][k]-(niEz[t-1][i+1][j][k]+niBy[t-1][i+1][j][k]
                   +niEz[t-1][i-1][j][k]-niBy[t-1][i-1][j][k])/2;
niBy[t+1][i][j][k]=(-Ez[t][i][j][k])+By[t][i][j][k]+Ez[t][i+1][j][k]
                   +By[t][i+1][j][k]-(niEz[t-1][i+1][j][k]+niBy[t-1][i+1][j][k]
                   -niEz[t-1][i-1][j][k]+niBy[t-1][i-1][j][k])/2;
skip2:;
/* half fields on top of the cube, above i,j,k    */
if (k==kmax-1) goto skip3;
nkEx[t+1][i][j][k]=Ex[t][i][j][k]+By[t][i][j][k]+Ex[t][i][j][k+1]
                   -By[t][i][j][k+1]-(nkEx[t-1][i][j][k+1]-nkBy[t-1][i][j][k+1]
                   +nkEx[t-1][i][j][k-1]+nkBy[t-1][i][j][k-1])/2;
```

```
nkBy[t+1][i][j][k]=Ex[t][i][j][k]+By[t][i][j][k]-Ex[t][i][j][k+1]
                  +By[t][i][j][k+1]-(-nkEx[t-1][i][j][k+1]+nkBy[t-1][i][j][k+1]
                  +nkEx[t-1][i][j][k-1]+nkBy[t-1][i][j][k-1])/2;
nkEy[t+1][i][j][k]=Ey[t][i][j][k]-Bx[t][i][j][k]+Ey[t][i][j][k+1]
                  +Bx[t][i][j][k+1]-(nkEy[t-1][i][j][k+1]+nkBx[t-1][i][j][k+1]
                  +nkEy[t-1][i][j][k-1]-nkBx[t-1][i][j][k-1])/2;
nkBx[t+1][i][j][k]=(-Ey[t][i][j][k])+Bx[t][i][j][k]+Ey[t][i][j][k+1]
                   +Bx[t][i][j][k+1]-(nkEy[t-1][i][j][k+1]+nkBx[t-1][i][j][k+1]
                   -nkEy[t-1][i][j][k-1]+nkBx[t-1][i][j][k-1])/2;
skip3:;
            }
       }
   }
   /* This is for fields that exist at the interface of   */
   /* the wave guide and dielectric regions.              */
   j=jmax-1;
   for (i=3;i<=(imax-3);++i)
   {   for (k=3;k<=(kmax-3);++k)
       {
njEx[t+1][i][j][k]=Ex[t][i][j][k]-Bz[t][i][j][k]+Exd[t][i][1][k]
                  +Bzd[t][i][1][k]-(njExd[t-1][i][1][k]
                  +njBzd[t-1][i][1][k]+njEx[t-1][i][j-1][k]
                  -njBz[t-1][i][j-1][k])/2;
njBz[t+1][i][j][k]=(-Ex[t][i][j][k])+Bz[t][i][j][k]+Exd[t][i][1][k]
                  +Bzd[t][i][1][k]-(njExd[t-1][i][1][k]
                  +njBzd[t-1][i][1][k]-njEx[t-1][i][j-1][k]
                  +njBz[t-1][i][j-1][k])/2;
njEz[t+1][i][j][k]=Ez[t][i][j][k]+Bx[t][i][j][k]+Ezd[t][i][1][k]
                  -Bxd[t][i][1][k]-(njEzd[t-1][i][1][k]
                  -njBxd[t-1][i][1][k]+njEz[t-1][i][j-1][k]
                  +njBx[t-1][i][j-1][k])/2;
njBx[t+1][i][j][k]=Ez[t][i][j][k]+Bx[t][i][j][k]-Ezd[t][i][1][k]
                  +Bxd[t][i][1][k]-(-njEzd[t-1][i][1][k]
                  +njBxd[t-1][i][1][k]+njEz[t-1][i][j-1][k]
                  +njBx[t-1][i][j-1][k])/2;
       }
   }
/*************************************************************/
/*  Calculate the node field components for the              */
/*  waveguide region for the next full time step (ie t+1).*/
/*************************************************************/
   for (k=1;k<=(kmax-1);++k)
   {   for (j=1;j<=(jmax-1);++j)
       {   for (i=1;i<=(imax-1);++i)
           {
Ex[t+2][i][j][k]=Ex[t][i][j][k]+(njBz[t+1][i][j][k]-njBz[t+1][i][j-1][k]
                 -nkBy[t+1][i][j][k]+nkBy[t+1][i][j][k-1])/er;
Ey[t+2][i][j][k]=Ey[t][i][j][k]+(nkBx[t+1][i][j][k]-nkBx[t+1][i][j][k-1]
                 -niBz[t+1][i][j][k]+niBz[t+1][i-1][j][k])/er;
               if (j==15) /* The plane of excitation  */
               {
```

```
Ez[t+2][i][j][k]=Ez[t][i][j][k]+(niBy[t+1][i][j][k]-niBy[t+1][i-1][j][k]
                -njBx[t+1][i][j][k]+njBx[t+1][i][j-1][k]+njBxexcit[i][k])/er;
Bx[t+2][i][j][k]=Bx[t][i][j][k]+(nkEy[t+1][i][j][k]-nkEy[t+1][i][j][k-1]
                -njEz[t+1][i][j][k]+njEz[t+1][i][j-1][k]+njEzexcit[i][k])/ur;
              }
              else
              {
Ez[t+2][i][j][k]=Ez[t][i][j][k]+(niBy[t+1][i][j][k]-niBy[t+1][i-1][j][k]
                -njBx[t+1][i][j][k]+njBx[t+1][i][j-1][k])/er;
Bx[t+2][i][j][k]=Bx[t][i][j][k]+(nkEy[t+1][i][j][k]-nkEy[t+1][i][j][k-1]
                -njEz[t+1][i][j][k]+njEz[t+1][i][j-1][k])/ur;
              }
By[t+2][i][j][k]=By[t][i][j][k]+(niEz[t+1][i][j][k]-niEz[t+1][i-1][j][k]
                -nkEx[t+1][i][j][k]+nkEx[t+1][i][j][k-1])/ur;
Bz[t+2][i][j][k]=Bz[t][i][j][k]+(njEx[t+1][i][j][k]-njEx[t+1][i][j-1][k]
                -niEy[t+1][i][j][k]+niEy[t+1][i-1][j][k])/ur;
          }
       }
   }
   /*****************************************************************/
   /* This section of the code calculates the field components  */
   /* for the dielectric slab which covers the end of the guide.*/
   /*****************************************************************/
   /* field components continuous across opening in the guide  */
   /* They are updated above.                                  */
    for (i=3;i<=(imax-3);++i)
    {   for (k=3;k<=(kmax-3);++k)
       {   njExd[t+1][i][0][k]=njEx[t+1][i][jmax-1][k];
           njBzd[t+1][i][0][k]=njBz[t+1][i][jmax-1][k];
           njEzd[t+1][i][0][k]=njEz[t+1][i][jmax-1][k];
           njBxd[t+1][i][0][k]=njBx[t+1][i][jmax-1][k];
       }
    }

   /*  E wall boundary to the left for dielectric region  */
   j=0;
   for (i=1;i<=2;++i)
   {   for (k=1;k<=(kmaxd-1);++k)
       {   njExd[t+1][i][j][k]=0.0;
           njBzd[t+1][i][j][k]=(2.0)*(Exd[t][i][j+1][k]+Bzd[t][i][j+1][k])
                               -(njExd[t-1][i][j+1][k]+njBzd[t-1][i][j+1][k]);
           njEzd[t+1][i][j][k]=0.0;
           njBxd[t+1][i][j][k]=(-2.0)*(Ezd[t][i][j+1][k]-Bxd[t][i][j+1][k])
                               +(njEzd[t-1][i][j+1][k]-njBxd[t-1][i][j+1][k]);
       }
   }
   for (i=imaxd-2;i<=imaxd-1;++i)
   {   for (k=1;k<=(kmaxd-1);++k)
       {   njExd[t+1][i][j][k]=0.0;
           njBzd[t+1][i][j][k]=(2.0)*(Exd[t][i][j+1][k]+Bzd[t][i][j+1][k])
                               -(njExd[t-1][i][j+1][k]+njBzd[t-1][i][j+1][k]);
           njEzd[t+1][i][j][k]=0.0;
```

```
        njBxd[t+1][i][j][k]=(-2.0)*(Ezd[t][i][j+1][k]-Bxd[t][i][j+1][k])
                            +(njEzd[t-1][i][j+1][k]-njBxd[t-1][i][j+1][k]);
    }
}
for (i=3;i<=imaxd-3;++i)
{   for (k=1;k<=2;++k)
    {   njExd[t+1][i][j][k]=0.0;
        njBzd[t+1][i][j][k]=(2.0)*(Exd[t][i][j+1][k]+Bzd[t][i][j+1][k])
                            -(njExd[t-1][i][j+1][k]+njBzd[t-1][i][j+1][k]);
        njEzd[t+1][i][j][k]=0.0;
        njBxd[t+1][i][j][k]=(-2.0)*(Ezd[t][i][j+1][k]-Bxd[t][i][j+1][k])
                            +(njEzd[t-1][i][j+1][k]-njBxd[t-1][i][j+1][k]);
    }
}
for (i=3;i<=imaxd-3;++i)
{   for (k=kmaxd-2;k<=(kmaxd-1);++k)
    {   njExd[t+1][i][j][k]=0.0;
        njBzd[t+1][i][j][k]=(2.0)*(Exd[t][i][j+1][k]+Bzd[t][i][j+1][k])
                            -(njExd[t-1][i][j+1][k]+njBzd[t-1][i][j+1][k]);
        njEzd[t+1][i][j][k]=0.0;
        njBxd[t+1][i][j][k]=(-2.0)*(Ezd[t][i][j+1][k]-Bxd[t][i][j+1][k])
                            +(njEzd[t-1][i][j+1][k]-njBxd[t-1][i][j+1][k]);
    }
}
j=1;
for (k=1;k<=(kmaxd-1);++k)
{   for (i=1;i<=(imaxd-1);++i)
    {   njExd[t+1][i][j][k]=Exd[t][i][j][k]-Bzd[t][i][j][k]+Exd[t][i][j+1][k]
                +Bzd[t][i][j+1][k]-(njExd[t-1][i][j+1][k]
                +njBzd[t-1][i][j+1][k]+njExd[t-1][i][j-1][k]
                -njBzd[t-1][i][j-1][k])/2;
        njBzd[t+1][i][j][k]=(-Exd[t][i][j][k])+Bzd[t][i][j][k]+Exd[t][i][j+1][k]
                +Bzd[t][i][j+1][k]-(njExd[t-1][i][j+1][k]
                +njBzd[t-1][i][j+1][k]-njExd[t-1][i][j-1][k]
                +njBzd[t-1][i][j-1][k])/2;
        njEzd[t+1][i][j][k]=Ezd[t][i][j][k]+Bxd[t][i][j][k]+Ezd[t][i][j+1][k]
                -Bxd[t][i][j+1][k]-(njEzd[t-1][i][j+1][k]
                -njBxd[t-1][i][j+1][k]+njEzd[t-1][i][j-1][k]
                +njBxd[t-1][i][j-1][k])/2;
        njBxd[t+1][i][j][k]=Ezd[t][i][j][k]+Bxd[t][i][j][k]-Ezd[t][i][j+1][k]
                +Bxd[t][i][j+1][k]-(-njEzd[t-1][i][j+1][k]
                +njBxd[t-1][i][j+1][k]+njEzd[t-1][i][j-1][k]
                +njBxd[t-1][i][j-1][k])/2;
    }
}
j=2;
for (k=1;k<=(kmaxd-1);++k)
{   for (i=1;i<=(imaxd-1);++i)
    {   njExd[t+1][i][j][k]=Exd[t][i][j][k]-Bzd[t][i][j][k]+Exf[t][i+10][10][k+16]
                +Bzf[t][i+10][10][k+16]-(njExf[t-1][i+10][10][k+16]
                +njBzf[t-1][i+10][10][k+16]+njExd[t-1][i][j-1][k]
```

```
                -njBzd[t-1][i][j-1][k])/2;
        njBzd[t+1][i][j][k]=(-Exd[t][i][j][k])+Bzd[t][i][j][k]+Exf[t][i+10][10][k+16]
                +Bzf[t][i+10][10][k+16]-(njExf[t-1][i+10][10][k+16]
                +njBzf[t-1][i+10][10][k+16]-njExd[t-1][i][j-1][k]
                +njBzd[t-1][i][j-1][k])/2;
        njEzd[t+1][i][j][k]=Ezd[t][i][j][k]+Bxd[t][i][j][k]+Ezf[t][i+10][ 10][k+16]
                -Bxf[t][i+10][10][k+16]-(njEzf[t-1][i+10][10][k+16]
                -njBxf[t-1][i+10][10][k+16]+njEzd[t-1][i][j-1][k]
                +njBxd[t-1][i][j-1][k])/2;
        njBxd[t+1][i][j][k]=Ezd[t][i][j][k]+Bxd[t][i][j][k]-Ezf[t][i+10][10][k+16]
                +Bxf[t][i+10][10][k+16]-(-njEzf[t-1][i+10][10][k+16]
                +njBxf[t-1][i+10][10][k+16]+njEzd[t-1][i][j-1][k]
                +njBxd[t-1][i][j-1][k])/2;
  }
}
i=0;
for (k=1;k<=(kmaxd-1);++k)
{   for (j=1;j<=(jmaxd-1);++j)
    {   niEyd[t+1][i][j][k]=Eyf[t][10][j+7][k+16]+Bzf[t][10][j+7][k+16]
                +Eyd[t][i+1][j][k]-Bzd[t][i+1][j][k]-(niEyd[t-1][i+1][j][k]
                -niBzd[t-1][i+1][j][k]+niEyf[t-1][9][j+7][k+16]
                +niBzf[t-1][9][j+7][k+16])/2;
        niBzd[t+1][i][j][k]=Eyf[t][10][j+7][k+16]+Bzf[t][10][j+7][k+16]
                -Eyd[t][i+1][j][k]+Bzd[t][i+1][j][k]-(-niEyd[t-1][i+1][j][k]
                +niBzd[t-1][i+1][j][k]+niEyf[t-1][9][j+7][k+16]
                +niBzf[t-1][9][j+7][k+16])/2;
        niEzd[t+1][i][j][k]=Ezf[t][10][j+7][k+16]-Byf[t][10][j+7][k+16]
                +Ezd[t][i+1][j][k]+Byd[t][i+1][j][k]-(niEzd[t-1][i+1][j][k]
                +niByd[t-1][i+1][j][k]+niEzf[t-1][9][j+7][k+16]
                -niByf[t-1][9][j+7][k+16])/2;
       niByd[t+1][i][j][k]=(-Ezf[t][10][j+7][k+16])+Byf[t][10][j+7][k+16]
                +Ezd[t][i+1][j][k]+Byd[t][i+1][j][k]-(niEzd[t-1][i+1][j][k]
                +niByd[t-1][i+1][j][k]-niEzf[t-1][9][j+7][k+16]
                +niByf[t-1][9][j+7][k+16])/2;
    }
}
for (k=1;k<=(kmaxd-1);++k)
{   for (j=1;j<=(jmaxd-1);++j)
    {   for (i=1;i<=(imaxd-2);++i)
        {   niEyd[t+1][i][j][k]=Eyd[t][i][j][k]+Bzd[t][i][j][k]+Eyd[t][i+1][j][k]
                    -Bzd[t][i+1][j][k]-(niEyd[t-1][i+1][j][k]
                    -niBzd[t-1][i+1][j][k]+niEyd[t-1][i-1][j][k]
                    +niBzd[t-1][i-1][j][k])/2;
            niBzd[t+1][i][j][k]=Eyd[t][i][j][k]+Bzd[t][i][j][k]-Eyd[t][i+1][j][k]
                    +Bzd[t][i+1][j][k]-(-niEyd[t-1][i+1][j][k]
                    +niBzd[t-1][i+1][j][k]+niEyd[t-1][i-1][j][k]
                    +niBzd[t-1][i-1][j][k])/2;
            niEzd[t+1][i][j][k]=Ezd[t][i][j][k]-Byd[t][i][j][k]+Ezd[t][i+1][j][k]
                    +Byd[t][i+1][j][k]-(niEzd[t-1][i+1][j][k]
                    +niByd[t-1][i+1][j][k]+niEzd[t-1][i-1][j][k]
                    -niByd[t-1][i-1][j][k])/2;
```

```
            niByd[t+1][j][i][k]=(-Ezd[t][i][j][k])+Byd[t][i][j][k]+Ezd[t][i+1][j][k]
                    +Byd[t][i+1][j][k]-(niEzd[t-1][i+1][j][k]
                    +niByd[t-1][i+1][j][k]-niEzd[t-1][i-1][j][k]
                    +niByd[t-1][i-1][j][k])/2;
        }
    }
}
i=imaxd-1;
for (k=1;k<=(kmaxd-1);++k)
{   for (j=1;j<=(jmaxd-1);++j)
    {   niEyd[t+1][i][j][k]=Eyd[t][i][j][k]+Bzd[t][i][j][k]+Eyf[t][34][j+7][k+16]
                -Bzf[t][34][j+7][k+16]-(niEyf[t-1][34][j+7][k+16]
                -niBzf[t-1][34][j+7][k+16]+niEyd[t-1][i-1][j][k]
                +niBzd[t-1][i-1][j][k])/2;
        niBzd[t+1][i][j][k]=Eyd[t][i][j][k]+Bzd[t][i][j][k]-Eyf[t][34][j+7][k+16]
                +Bzf[t][34][j+7][k+16]-(-niEyf[t-1][34][j+7][k+16]
                +niBzf[t-1][34][j+7][k+16]+niEyd[t-1][i-1][j][k]
                +niBzd[t-1][i-1][j][k])/2;
       niEzd[t+1][i][j][k]=Ezd[t][i][j][k]-Byd[t][i][j][k]+Ezf[t][34][j+7][k+16]
                +Byf[t][34][j+7][k+16]-(niEzf[t-1][34][j+7][k+16]
                +niByf[t-1][34][j+7][k+16]+niEzd[t-1][i-1][j][k]
                -niByd[t-1][i-1][j][k])/2;
       niByd[t+1][i][j][k]=(-Ezd[t][i][j][k])+Byd[t][i][j][k]+Ezf[t][34][j+7][k+16]
                +Byf[t][34][j+7][k+16]-(niEzf[t-1][34][j+7][k+16]
                +niByf[t-1][34][j+7][k+16]-niEzd[t-1][i-1][j][k]
                +niByd[t-1][i-1][j][k])/2;
    }
}
k=0;
for (i=1;i<=(imaxd-1);++i)
{   for (j=1;j<=(jmaxd-1);++j)
    {    nkExd[t+1][i][j][k]=Exf[t][i+10][j+7][16]+Byf[t][i+10][j+7][16]
                +Exd[t][i][j][k+1]-Byd[t][i][j][k+1]-(nkExd[t-1][i][j][k+1]
                -nkByd[t-1][i][j][k+1]+nkExf[t-1][i+10][j+7][15]
                +nkByf[t-1][i+10][j+7][15])/2;
         nkByd[t+1][i][j][k]=Exf[t][i+10][j+7][16]+Byf[t][i+10][j+7][16]
                -Exd[t][i][j][k+1]+Byd[t][i][j][k+1]-(-nkExd[t-1][i][j][k+1]
                +nkByd[t-1][i][j][k+1]+nkExf[t-1][i+10][j+7][15]
                +nkByf[t-1][i+10][j+7][15])/2;
         nkEyd[t+1][i][j][k]=Eyf[t][i+10][j+7][16]-Bxf[t][i+10][j+7][16]
                +Eyd[t][i][j][k+1]+Bxd[t][i][j][k+1]-(nkEyd[t-1][i][j][k+1]
                +nkBxd[t-1][i][j][k+1]+nkEyf[t-1][i+10][j+7][15]
                -nkBxf[t-1][i+10][j+7][15])/2;
        nkBxd[t+1][i][j][k]=(-Eyf[t][i+10][j+7][16])+Bxf[t][i+10][j+7][16]
                +Eyd[t][i][j][k+1]+Bxd[t][i][j][k+1]-(nkEyd[t-1][i][j][k+1]
                +nkBxd[t-1][i][j][k+1]-nkEyf[t-1][i+10][j+7][15]
                +nkBxf[t-1][i+10][j+7][15])/2;
    }
}
for (k=1;k<=(kmaxd-2);++k)
{   for (j=1;j<=(jmaxd-1);++j)
```

```
    {   for (i=1;i<=(imaxd-1);++i)
        {   nkExd[t+1][i][j][k]=Exd[t][i][j][k]+Byd[t][i][j][k]+Exd[t][i][j][k+1]
                -Byd[t][i][j][k+1]-(nkExd[t-1][i][j][k+1]
                -nkByd[t-1][i][j][k+1]+nkExd[t-1][i][j][k-1]
                +nkByd[t-1][i][j][k-1])/2;
            nkByd[t+1][i][j][k]=Exd[t][i][j][k]+Byd[t][i][j][k]-Exd[t][i][j][k+1]
                +Byd[t][i][j][k+1]-(-nkExd[t-1][i][j][k+1]
                +nkByd[t-1][i][j][k+1]+nkExd[t-1][i][j][k-1]
                +nkByd[t-1][i][j][k-1])/2;
            nkEyd[t+1][i][j][k]=Eyd[t][i][j][k]-Bxd[t][i][j][k]+Eyd[t][i][j][k+1]
                +Bxd[t][i][j][k+1]-(nkEyd[t-1][i][j][k+1]
                +nkBxd[t-1][i][j][k+1]+nkEyd[t-1][i][j][k-1]
                -nkBxd[t-1][i][j][k-1])/2;
            nkBxd[t+1][i][j][k]=(-Eyd[t][i][j][k])+Bxd[t][i][j][k]+Eyd[t][i][j][k+1]
                +Bxd[t][i][j][k+1]-(nkEyd[t-1][i][j][k+1]
                +nkBxd[t-1][i][j][k+1]-nkEyd[t-1][i][j][k-1]
                +nkBxd[t-1][i][j][k-1])/2;
        }
    }
  }
  k=kmaxd-1;
  for (i=1;i<=(imaxd-1);++i)
  {   for (j=1;j<=(jmaxd-1);++j)
      {   nkExd[t+1][i][j][k]=Exd[t][i][j][k]+Byd[t][i][j][k]+Exf[t][i+10][j+7][27]
              -Byf[t][i+10][j+7][27]-(nkExf[t-1][i+10][j+7][27]
              -nkByf[t-1][i+10][j+7][27]+nkExd[t-1][i][j][k-1]
              +nkByd[t-1][i][j][k-1])/2;
          nkByd[t+1][i][j][k]=Exd[t][i][j][k]+Byd[t][i][j][k]-Exf[t][i+10][j+7][27]
              +Byf[t][i+10][j+7][27]-(-nkExf[t-1][i+10][j+7][27]
              +nkByf[t-1][i+10][j+7][27]+nkExd[t-1][i][j][k-1]
              +nkByd[t-1][i][j][k-1])/2;
          nkEyd[t+1][i][j][k]=Eyd[t][i][j][k]-Bxd[t][i][j][k]+Eyf[t][i+10][j+7][27]
              +Bxf[t][i+10][j+7][27]-(nkEyf[t-1][i+10][j+7][27]
              +nkBxf[t-1][i+10][j+7][27]+nkEyd[t-1][i][j][k-1]
              -nkBxd[t-1][i][j][k-1])/2;
          nkBxd[t+1][i][j][k]=(-Eyd[t][i][j][k])+Bxd[t][i][j][k]+Eyf[t][i+10][j+7][27]
              +Bxf[t][i+10][j+7][27]-(nkEyf[t-1][i+10][j+7][27]
              +nkBxf[t-1][i+10][j+7][27]-nkEyd[t-1][i][j][k-1]
              +nkBxd[t-1][i][j][k-1])/2;
      }
  }

/*************************************************************/
/*  Calculate the node field components for the              */
/*  dielectric region for the next full time step (ie t+1)*/
/*************************************************************/
  for (k=1;k<=(kmaxd-1);++k)
  {   for (j=1;j<=(jmaxd-1);++j)
      {   for (i=1;i<=(imaxd-1);++i)
          {
   Exd[t+2][i][j][k]=Exd[t][i][j][k]+(njBzd[t+1][i][j][k]-njBzd[t+1][i][j-1][k]
```

```
                    -nkByd[t+1][i][j][k]+nkByd[t+1][i][j][k-1])/erd;
 Eyd[t+2][i][j][k]=Eyd[t][i][j][k]+(nkBxd[t+1][i][j][k]-nkBxd[t+1][i][j][k-1]
                    -niBzd[t+1][i][j][k]+niBzd[t+1][i-1][j][k])/erd;
 Ezd[t+2][i][j][k]=Ezd[t][i][j][k]+(niByd[t+1][i][j][k]-niByd[t+1][i-1][j][k]
                    -njBxd[t+1][i][j][k]+njBxd[t+1][i][j-1][k])/erd;
 Bxd[t+2][i][j][k]=Bxd[t][i][j][k]+(nkEyd[t+1][i][j][k]-nkEyd[t+1][i][j][k-1]
                    -njEzd[t+1][i][j][k]+njEzd[t+1][i][j-1][k])/urd;
 Byd[t+2][i][j][k]=Byd[t][i][j][k]+(niEzd[t+1][i][j][k]-niEzd[t+1][i-1][j][k]
                    -nkExd[t+1][i][j][k]+nkExd[t+1][i][j][k-1])/urd;
 Bzd[t+2][i][j][k]=Bzd[t][i][j][k]+(njExd[t+1][i][j][k]-njExd[t+1][i][j-1][k]
                    -niEyd[t+1][i][j][k]+niEyd[t+1][i-1][j][k])/urd;
        }
    }
}
/*************************************************************/
/* This section of code deals with the free space region.    */
/* The half field components are first made to be            */
/* continuous, then half field components on the E walls     */
/* and absorbing boundaries are determined.  Finally,        */
/* half fieldsthroughout the region are calulated.           */
/*************************************************************/

/* Make half field components continous across the       */
/* boundary between the dielectric and free space regions*/
k=26;
for (j=8;j<=9;++j)
{   for (i=11;i<=33;++i)
    {   nkExf[t+1][i][j][k]=nkExd[t+1][i-10][j-7][k-16];
        nkByf[t+1][i][j][k]=nkByd[t+1][i-10][j-7][k-16];
        nkEyf[t+1][i][j][k]=nkEyd[t+1][i-10][j-7][k-16];
        nkBxf[t+1][i][j][k]=nkBxd[t+1][i-10][j-7][k-16];
    }
}
k=16;
for (j=8;j<=9;++j)
{   for (i=11;i<=33;++i)
    {   nkExf[t+1][i][j][k]=nkExd[t+1][i-10][j-7][k-16];
        nkByf[t+1][i][j][k]=nkByd[t+1][i-10][j-7][k-16];
        nkEyf[t+1][i][j][k]=nkEyd[t+1][i-10][j-7][k-16];
        nkBxf[t+1][i][j][k]=nkBxd[t+1][i-10][j-7][k-16];
    }
}
i=10;
for (j=8;j<=9;++j)
{   for (k=17;k<=26;++k)
    {   niEyf[t+1][i][j][k]=niEyd[t+1][i-10][j-7][k-16];
        niBzf[t+1][i][j][k]=niBzd[t+1][i-10][j-7][k-16];
        niEzf[t+1][i][j][k]=niEzd[t+1][i-10][j-7][k-16];
        niByf[t+1][i][j][k]=niByd[t+1][i-10][j-7][k-16];
    }
}
```

```
i=33;
for (j=8;j<=9;++j)
{   for (k=17;k<=26;++k)
    {   niEyf[t+1][i][j][k]=niEyd[t+1][i-10][j-7][k-16];
        niBzf[t+1][i][j][k]=niBzd[t+1][i-10][j-7][k-16];
        niEzf[t+1][i][j][k]=niEzd[t+1][i-10][j-7][k-16];
        niByf[t+1][i][j][k]=niByd[t+1][i-10][j-7][k-16];
    }
}
j=9;
for (i=11;i<=33;++i)
{   for (k=17;k<=26;++k)
    {   njExf[t+1][i][j][k]=njExd[t+1][i-10][j-7][k-16];
        njBxf[t+1][i][j][k]=njBxd[t+1][i-10][j-7][k-16];
        njEzf[t+1][i][j][k]=njEzd[t+1][i-10][j-7][k-16];
        njBzf[t+1][i][j][k]=njBzd[t+1][i-10][j-7][k-16];
    }
}

/* E wall sections in free space region  */
k=27;
for (j=0;j<=7;++j)
{   for (i=10;i<=34;++i)
    {   nkExf[t+1][i][j][k]=0.0;
        nkByf[t+1][i][j][k]=(-2.0)*(Exf[t][i][j][k+1]-Byf[t][i][j][k+1])
                            +(nkExf[t-1][i][j][k+1]-nkByf[t-1][i][j][k+1]);
        nkEyf[t+1][i][j][k]=0.0;
        nkBxf[t+1][i][j][k]=(2.0)*(Eyf[t][i][j][k+1]+Bxf[t][i][j][k+1])
                            -(nkEyf[t-1][i][j][k+1]+nkBxf[t-1][i][j][k+1]);
    }
}
k=15;
for (j=0;j<=7;++j)
{   for (i=10;i<=34;++i)
    {   nkExf[t+1][i][j][k]=0.0;
        nkByf[t+1][i][j][k]=(2.0)*(Exf[t][i][j][k]+Byf[t][i][j][k])
                            -(nkExf[t-1][i][j][k-1]+nkByf[t-1][i][j][k-1]);
        nkEyf[t+1][i][j][k]=0.0;
        nkBxf[t+1][i][j][k]=(-2.0)*(Eyf[t][i][j][k]-Bxf[t][i][j][k])
                            +(nkEyf[t-1][i][j][k-1]-nkBxf[t-1][i][j][k-1]);
    }
}
i=9;
for (j=0;j<=7;++j)
{   for (k=16;k<=27;++k)
    {   niEyf[t+1][i][j][k]=0.0;
        niBzf[t+1][i][j][k]=(2.0)*(Eyf[t][i][j][k]+Bzf[t][i][j][k])
                            -(niEyf[t-1][i-1][j][k]+niBzf[t-1][i-1][j][k]);
        niEzf[t+1][i][j][k]=0.0;
        niByf[t+1][i][j][k]=(-2.0)*(Ezf[t][i][j][k]-Byf[t][i][j][k])
                            +(niEzf[t-1][i-1][j][k]-niByf[t-1][i-1][j][k]);
```

```
    }
}
i=34;
for (j=0;j<=7;++j)
{   for (k=16;k<=27;++k)
    {   niEyf[t+1][i][j][k]=0.0;
        niBzf[t+1][i][j][k]=(-2.0)*(Eyf[t][i+1][j][k]-Bzf[t][i+1][j][k])
                            +(niEyf[t-1][i+1][j][k]-niBzf[t-1][i+1][j][k]);
        niEzf[t+1][i][j][k]=0.0;
        niByf[t+1][i][j][k]=(2.0)*(Ezf[t][i+1][j][k]+Byf[t][i+1][j][k])
                            -(niEzf[t-1][i+1][j][k]+niByf[t-1][i+1][j][k]);
    }
}
/*half field expressions for the lip */
/* ie account for thickness of guide */
j=7;
i=10;
for (k=16;k<=27;++k)
{   njExf[t+1][i][j][k]=0.0;
    njBzf[t+1][i][j][k]=(2.0)*(Exf[t][i][j+1][k]+Bzf[t][i][j+1][k])
                        -(njExf[t-1][i][j+1][k]+njBzf[t-1][i][j+1][k]);
    njEzf[t+1][i][j][k]=0.0;
    njBxf[t+1][i][j][k]=(-2.0)*(Ezf[t][i][j+1][k]-Bxf[t][i][j+1][k])
                        +(njEzf[t-1][i][j+1][k]-njBxf[t-1][i][j+1][k]);
}
j=7;
i=34;
for (k=16;k<=27;++k)
{   njExf[t+1][i][j][k]=0.0;
    njBzf[t+1][i][j][k]=(2.0)*(Exf[t][i][j+1][k]+Bzf[t][i][j+1][k])
                        -(njExf[t-1][i][j+1][k]+njBzf[t-1][i][j+1][k]);
    njEzf[t+1][i][j][k]=0.0;
    njBxf[t+1][i][j][k]=(-2.0)*(Ezf[t][i][j+1][k]-Bxf[t][i][j+1][k])
                        +(njEzf[t-1][i][j+1][k]-njBxf[t-1][i][j+1][k]);
}
j=7;
k=16;
for (i=10;i<=34;++i)
{   njExf[t+1][i][j][k]=0.0;
    njBzf[t+1][i][j][k]=(2.0)*(Exf[t][i][j+1][k]+Bzf[t][i][j+1][k])
                        -(njExf[t-1][i][j+1][k]+njBzf[t-1][i][j+1][k]);
    njEzf[t+1][i][j][k]=0.0;
    njBxf[t+1][i][j][k]=(-2.0)*(Ezf[t][i][j+1][k]-Bxf[t][i][j+1][k])
                        +(njEzf[t-1][i][j+1][k]-njBxf[t-1][i][j+1][k]);
}
j=7;
k=27;
for (i=10;i<=34;++i)
{   njExf[t+1][i][j][k]=0.0;
    njBzf[t+1][i][j][k]=(2.0)*(Exf[t][i][j+1][k]+Bzf[t][i][j+1][k])
                        -(njExf[t-1][i][j+1][k]+njBzf[t-1][i][j+1][k]);
```

```
    njEzf[t+1][i][j][k]=0.0;
    njBxf[t+1][i][j][k]=(-2.0)*(Ezf[t][i][j+1][k]-Bxf[t][i][j+1][k])
                        +(njEzf[t-1][i][j+1][k]-njBxf[t-1][i][j+1][k]);
}
/* Absorbing boundary in free space to the right   */
j=jmaxf-1;
for (i=1;i<=(imaxf-1);++i)
{   for (k=1;k<=(kmaxf-1);++k)
    {    njExf[t+1][i][j][k]=Exf[t][i][j][k]-Bzf[t][i][j][k]
                    -0.5*(njExf[t-1][i][j-1][k]-njBzf[t-1][i][j-1][k])
                    +1.125*(njE2xf[n-4][i][2][k]+njB2zf[n-4][i][2][k])
                    -0.75*(njE2xf[n-8][i][1][k]+njB2zf[n-8][i][1][k])
                    +0.125*(njE2xf[n-12][i][0][k]+njB2zf[n-12][i][0][k]);
        njBzf[t+1][i][j][k]=(-1.0)*Exf[t][i][j][k]+Bzf[t][i][j][k]
                    +0.5*(njExf[t-1][i][j-1][k]-njBzf[t-1][i][j-1][k])
                    +1.125*(njE2xf[n-4][i][2][k]+njB2zf[n-4][i][2][k])
                    -0.75*(njE2xf[n-8][i][1][k]+njB2zf[n-8][i][1][k])
                    +0.125*(njE2xf[n-12][i][0][k]+njB2zf[n-12][i][0][k]);
        njEzf[t+1][i][j][k]=Ezf[t][i][j][k]+Bxf[t][i][j][k]
                    -0.5*(njEzf[t-1][i][j-1][k]+njBxf[t-1][i][j-1][k])
                    +1.125*(njE2zf[n-4][i][2][k]-njB2xf[n-4][i][2][k])
                    -0.75*(njE2zf[n-8][i][1][k]-njB2xf[n-8][i][1][k])
                    +0.125*(njE2zf[n-12][i][0][k]-njB2xf[n-12][i][0][k]);
        njBxf[t+1][i][j][k]=Ezf[t][i][j][k]+Bxf[t][i][j][k]
                    -0.5*(njEzf[t-1][i][j-1][k]+njBxf[t-1][i][j-1][k])
                    -1.125*(njE2zf[n-4][i][2][k]-njB2xf[n-4][i][2][k])
                    +0.75*(njE2zf[n-8][i][1][k]-njB2xf[n-8][i][1][k])
                    -0.125*(njE2zf[n-12][i][0][k]-njB2xf[n-12][i][0][k]);
    }
}
/* Absorbing wall boundary to the left */
j=0;
for (i=1;i<=(imaxf-1);++i)
{   for (k=1;k<=(kmaxf-1);++k)
    {   if (((i<=34)&&(i>=10))&&((k<=27)&&(k>=16))) goto skip7;
        njExf[t+1][i][j][k]=Exf[t][i][j+1][k]+Bzf[t][i][j+1][k]

                    -0.5*(njExf[t-1][i][j+1][k]+njBzf[t-1][i][j+1][k])
                    +1.125*(njE1xf[n-4][i][0][k]-njB1zf[n-4][i][0][k])
                    -0.75*(njE1xf[n-8][i][1][k]-njB1zf[n-8][i][1][k])
                    +0.125*(njE1xf[n-12][i][2][k]-njB1zf[n-12][i][2][k]);
        njBzf[t+1][i][j][k]=Exf[t][i][j+1][k]+Bzf[t][i][j+1][k]
                    -0.5*(njExf[t-1][i][j+1][k]+njBzf[t-1][i][j+1][k])
                    -1.125*(njE1xf[n-4][i][0][k]-njB1zf[n-4][i][0][k])
                    +0.75*(njE1xf[n-8][i][1][k]-njB1zf[n-8][i][1][k])
                    -0.125*(njE1xf[n-12][i][2][k]-njB1zf[n-12][i][2][k]);
        njEzf[t+1][i][j][k]=Ezf[t][i][j+1][k]-Bxf[t][i][j+1][k]
                    -0.5*(njEzf[t-1][i][j+1][k]-njBxf[t-1][i][j+1][k])
                    +1.125*(njE1zf[n-4][i][0][k]+njB1xf[n-4][i][0][k])
                    -0.75*(njE1zf[n-8][i][1][k]+njB1xf[n-8][i][1][k])
                    +0.125*(njE1zf[n-12][i][2][k]+njB1xf[n-12][i][2][k]);
        njBxf[t+1][i][j][k]=(-1.0)*Ezf[t][i][j+1][k]+Bxf[t][i][j+1][k]
```

```
                    +0.5*(njEzf[t-1][i][j+1][k]-njBxf[t-1][i][j+1][k])
                    +1.125*(njE1zf[n-4][i][0][k]+njB1xf[n-4][i][0][k])
                    -0.75*(njE1zf[n-8][i][1][k]+njB1xf[n-8][i][1][k])
                    +0.125*(njE1zf[n-12][i][2][k]+njB1xf[n-12][i][2][k]);
        skip7:;
    }
}
/* Absorbing boundary on top  */
k=kmaxf-1;
for (i=1;i<=(imaxf-1);++i)
{   for (j=1;j<=(jmaxf-1);++j)
    {   nkEyf[t+1][i][j][k]=Eyf[t][i][j][k]-Bxf[t][i][j][k]
                    -0.5*(nkEyf[t-1][i][j][k-1]-nkBxf[t-1][i][j][k-1])
                    +1.125*(nkE2yf[n-4][i][j][2]+nkB2xf[n-4][i][j][2])
                    -0.75*(nkE2yf[n-8][i][j][1]+nkB2xf[n-8][i][j][1])
                    +0.125*(nkE2yf[n-12][i][j][0]+nkB2xf[n-12][i][j][0]);
        nkBxf[t+1][i][j][k]=(-1.0)*Eyf[t][i][j][k]+Bxf[t][i][j][k]
                    +0.5*(nkEyf[t-1][i][j][k-1]-nkBxf[t-1][i][j][k-1])
                    +1.125*(nkE2yf[n-4][i][j][2]+nkB2xf[n-4][i][j][2])
                    -0.75*(nkE2yf[n-8][i][j][1]+nkB2xf[n-8][i][j][1])
                    +0.125*(nkE2yf[n-12][i][j][0]+nkB2xf[n-12][i][j][0]);
        nkExf[t+1][i][j][k]=Exf[t][i][j][k]+Byf[t][i][j][k]
                    -0.5*(nkExf[t-1][i][j][k-1]+nkByf[t-1][i][j][k-1])
                    +1.125*(nkE2xf[n-4][i][j][2]-nkB2yf[n-4][i][j][2])
                    -0.75*(nkE2xf[n-8][i][j][1]-nkB2yf[n-8][i][j][1])
                    +0.125*(nkE2xf[n-12][i][j][0]-nkB2yf[n-12][i][j][0]);
        nkByf[t+1][i][j][k]=Exf[t][i][j][k]+Byf[t][i][j][k]
                    -0.5*(nkExf[t-1][i][j][k-1]+nkByf[t-1][i][j][k-1])
                    -1.125*(nkE2xf[n-4][i][j][2]-nkB2yf[n-4][i][j][2])
                    +0.75*(nkE2xf[n-8][i][j][1]-nkB2yf[n-8][i][j][1])
                    -0.125*(nkE2xf[n-12][i][j][0]-nkB2yf[n-12][i][j][0]);
    }
}
/* Absorbing boundary on the bottom  */
k=0;
for (i=1;i<=(imaxf-1);++i)
{   for (j=1;j<=(jmaxf-1);++j)
    {   nkEyf[t+1][i][j][k]=Eyf[t][i][j][k+1]+Bxf[t][i][j][k+1]
                    -0.5*(nkEyf[t-1][i][j][k+1]+nkBxf[t-1][i][j][k+1])
                    +1.125*(nkE1yf[n-4][i][j][0]-nkB1xf[n-4][i][j][0])
                    -0.75*(nkE1yf[n-8][i][j][1]-nkB1xf[n-8][i][j][1])
                    +0.125*(nkE1yf[n-12][i][j][2]-nkB1xf[n-12][i][j][2]);
        nkBxf[t+1][i][j][k]=Eyf[t][i][j][k+1]+Bxf[t][i][j][k+1]
                    -0.5*(nkEyf[t-1][i][j][k+1]+nkBxf[t-1][i][j][k+1])
                    -1.125*(nkE1yf[n-4][i][j][0]-nkB1xf[n-4][i][j][0])
                    +0.75*(nkE1yf[n-8][i][j][1]-nkB1xf[n-8][i][j][1])
                    -0.125*(nkE1yf[n-12][i][j][2]-nkB1xf[n-12][i][j][2]);
        nkExf[t+1][i][j][k]=Exf[t][i][j][k+1]-Byf[t][i][j][k+1]
                    -0.5*(nkExf[t-1][i][j][k+1]-nkByf[t-1][i][j][k+1])
                    +1.125*(nkE1xf[n-4][i][j][0]+nkB1yf[n-4][i][j][0])
                    -0.75*(nkE1xf[n-8][i][j][1]+nkB1yf[n-8][i][j][1])
```

```
                    +0.125*(nkE1xf[n-12][i][j][2]+nkB1yf[n-12][i][j][2]);
        nkByf[t+1][i][j][k]=(-1.0)*Exf[t][i][j][k+1]+Byf[t][i][j][k+1]
                    +0.5*(nkExf[t-1][i][j][k+1]-nkByf[t-1][i][j][k+1])
                    +1.125*(nkE1xf[n-4][i][j][0]+nkB1yf[n-4][i][j][0])
                    -0.75*(nkE1xf[n-8][i][j][1]+nkB1yf[n-8][i][j][1])
                    +0.125*(nkE1xf[n-12][i][j][2]+nkB1yf[n-12][i][j][2]);
    }
}
/* Absorbing boundary in front    */
i=imaxf-1;
for (j=1;j<=(jmaxf-1);++j)
{   for (k=1;k<=(kmaxf-1);++k)
    {   niEzf[t+1][i][j][k]=Ezf[t][i][j][k]-Byf[t][i][j][k]
                    -0.5*(niEzf[t-1][i-1][j][k]-niByf[t-1][i-1][j][k])
                    +1.125*(niE2zf[n-4][2][j][k]+niB2yf[n-4][2][j][k])
                    -0.75*(niE2zf[n-8][1][j][k]+niB2yf[n-8][1][j][k])
                    +0.125*(niE2zf[n-12][0][j][k]+niB2yf[n-12][0][j][k]);
        niByf[t+1][i][j][k]=(-1.0)*Ezf[t][i][j][k]+Byf[t][i][j][k]
                    +0.5*(niEzf[t-1][i-1][j][k]-niByf[t-1][i-1][j][k])
                    +1.125*(niE2zf[n-4][2][j][k]+niB2yf[n-4][2][j][k])
                    -0.75*(niE2zf[n-8][1][j][k]+niB2yf[n-8][1][j][k])
                    +0.125*(niE2zf[n-12][0][j][k]+niB2yf[n-12][0][j][k]);
        niEyf[t+1][i][j][k]=Eyf[t][i][j][k]+Bzf[t][i][j][k]
                    -0.5*(niEyf[t-1][i-1][j][k]+niBzf[t-1][i-1][j][k])
                    +1.125*(niE2yf[n-4][2][j][k]-niB2zf[n-4][2][j][k])
                    -0.75*(niE2yf[n-8][1][j][k]-niB2zf[n-8][1][j][k])
                    +0.125*(niE2yf[n-12][0][j][k]-niB2zf[n-12][0][j][k]);
        niBzf[t+1][i][j][k]=Eyf[t][i][j][k]+Bzf[t][i][j][k]
                    -0.5*(niEyf[t-1][i-1][j][k]+niBzf[t-1][i-1][j][k])
                    -1.125*(niE2yf[n-4][2][j][k]-niB2zf[n-4][2][j][k])
                    +0.75*(niE2yf[n-8][1][j][k]-niB2zf[n-8][1][j][k])
                    -0.125*(niE2yf[n-12][0][j][k]-niB2zf[n-12][0][j][k]);
    }
}
/* Absorbing wall boundary in the back */
i=0;
for (j=1;j<=(jmaxf-1);++j)
{   for (k=1;k<=(kmaxf-1);++k)
    {   niEzf[t+1][i][j][k]=Ezf[t][i+1][j][k]+Byf[t][i+1][j][k]
                    -0.5*(niEzf[t-1][i+1][j][k]+niByf[t-1][i+1][j][k])
                    +1.125*(niE1zf[n-4][0][j][k]-niB1yf[n-4][0][j][k])
                    -0.75*(niE1zf[n-8][1][j][k]-niB1yf[n-8][1][j][k])
                    +0.125*(niE1zf[n-12][2][j][k]-niB1yf[n-12][2][j][k]);
        niByf[t+1][i][j][k]=Ezf[t][i+1][j][k]+Byf[t][i+1][j][k]
                    -0.5*(niEzf[t-1][i+1][j][k]+niByf[t-1][i+1][j][k])
                    -1.125*(niE1zf[n-4][0][j][k]-niB1yf[n-4][0][j][k])
                    +0.75*(niE1zf[n-8][1][j][k]-niB1yf[n-8][1][j][k])
                    -0.125*(niE1zf[n-12][2][j][k]-niB1yf[n-12][2][j][k]);
        niEyf[t+1][i][j][k]=Eyf[t][i+1][j][k]-Bzf[t][i+1][j][k]
                    -0.5*(niEyf[t-1][i+1][j][k]-niBzf[t-1][i+1][j][k])
                    +1.125*(niE1yf[n-4][0][j][k]+niB1zf[n-4][0][j][k])
```

```
                        -0.75*(niE1yf[n-8][1][j][k]+niB1zf[n-8][1][j][k])
                        +0.125*(niE1yf[n-12][2][j][k]+niB1zf[n-12][2][j][k]);
            niBzf[t+1][i][j][k]=(-1.0)*Eyf[t][i+1][j][k]+Bzf[t][i+1][j][k]
                        +0.5*(niEyf[t-1][i+1][j][k]-niBzf[t-1][i+1][j][k])
                        +1.125*(niE1yf[n-4][0][j][k]+niB1zf[n-4][0][j][k])
                        -0.75*(niE1yf[n-8][1][j][k]+niB1zf[n-8][1][j][k])
                        +0.125*(niE1yf[n-12][2][j][k]+niB1zf[n-12][2][j][k]);
        }
    }
    /* Calculate half fields for next time step in free space region  */
    for (i=1;i<=imaxf-1;++i)
    {   for (j=1;j<=jmaxf-1;++j)
        {   for (k=1;k<=kmaxf-1;++k)
            {
if (((k>=16)&&(k<=27))&&((i>=10)&&(i<=34))&&(j<=7)) goto skip8;
if (((k>=17)&&(k<=26))&&((i>=11)&&(i<=33))&&((j==8)||(j==9))) goto skip8;
if (j==jmaxf-1) goto skip8;
                njExf[t+1][i][j][k]=Exf[t][i][j][k]-Bzf[t][i][j][k]+Exf[t][i][j+1][k]
                    +Bzf[t][i][j+1][k]-(njExf[t-1][i][j+1][k]
                    +njBzf[t-1][i][j+1][k]+njExf[t-1][i][j-1][k]
                    -njBzf[t-1][i][j-1][k])/2;
                njBzf[t+1][i][j][k]=(-Exf[t][i][j][k])+Bzf[t][i][j][k]+Exf[t][i][j+1][k]
                    +Bzf[t][i][j+1][k]-(njExf[t-1][i][j+1][k]
                    +njBzf[t-1][i][j+1][k]-njExf[t-1][i][j-1][k]
                    +njBzf[t-1][i][j-1][k])/2;
                njEzf[t+1][i][j][k]=Ezf[t][i][j][k]+Bxf[t][i][j][k]+Ezf[t][i][j+1][k]
                    -Bxf[t][i][j+1][k]-(njEzf[t-1][i][j+1][k]
                    -njBxf[t-1][i][j+1][k]+njEzf[t-1][i][j-1][k]
                    +njBxf[t-1][i][j-1][k])/2;
                njBxf[t+1][i][j][k]=Ezf[t][i][j][k]+Bxf[t][i][j][k]-Ezf[t][i][j+1][k]
                    +Bxf[t][i][j+1][k]-(-njEzf[t-1][i][j+1][k]
                    +njBxf[t-1][i][j+1][k]+njEzf[t-1][i][j-1][k]
                    +njBxf[t-1][i][j-1][k])/2;
skip8:;
if (((k>=16)&&(k<=27))&&((i>=9)&&(i<=34))&&(j<=7)) goto skip9;
if (((k>=17)&&(k<=26))&&((i>=10)&&(i<=33))&&((j==8)||(j==9))) goto skip9;
if (i==imaxf-1) goto skip9;
                niEyf[t+1][i][j][k]=Eyf[t][i][j][k]+Bzf[t][i][j][k]+Eyf[t][i+1][j][k]
                    -Bzf[t][i+1][j][k]-(niEyf[t-1][i+1][j][k]
                    -niBzf[t-1][i+1][j][k]+niEyf[t-1][i-1][j][k]
                    +niBzf[t-1][i-1][j][k])/2;
                niBzf[t+1][i][j][k]=Eyf[t][i][j][k]+Bzf[t][i][j][k]-Eyf[t][i+1][j][k]
                    +Bzf[t][i+1][j][k]-(-niEyf[t-1][i+1][j][k]
                    +niBzf[t-1][i+1][j][k]+niEyf[t-1][i-1][j][k]
                    +niBzf[t-1][i-1][j][k])/2;
                niEzf[t+1][i][j][k]=Ezf[t][i][j][k]-Byf[t][i][j][k]+Ezf[t][i+1][j][k]
                    +Byf[t][i+1][j][k]-(niEzf[t-1][i+1][j][k]
                    +niByf[t-1][i+1][j][k]+niEzf[t-1][i-1][j][k]
                    -niByf[t-1][i-1][j][k])/2;
                niByf[t+1][i][j][k]=(-Ezf[t][i][j][k])+Byf[t][i][j][k]+Ezf[t][i+1][j][k]
                    +Byf[t][i+1][j][k]-(niEzf[t-1][i+1][j][k]
```

```
                    +niByf[t-1][i+1][j][k]-niEzf[t-1][i-1][j][k]
                    +niByf[t-1][i-1][j][k])/2;
skip9:;
if (((k>=15)&&(k<=27))&&((i>=10)&&(i<=34))&&(j<=7)) goto skip10;
if (((k>=16)&&(k<=26))&&((i>=11)&&(i<=33))&&((j==8)||(j==9))) goto skip10;
if (k==kmaxf-1) goto skip10;
              nkExf[t+1][i][j][k]=Exf[t][i][j][k]+Byf[t][i][j][k]+Exf[t][i][j][k+1]
                    -Byf[t][i][j][k+1]-(nkExf[t-1][i][j][k+1]
                    -nkByf[t-1][i][j][k+1]+nkExf[t-1][i][j][k-1]
                    +nkByf[t-1][i][j][k-1])/2;
              nkByf[t+1][i][j][k]=Exf[t][i][j][k]+Byf[t][i][j][k]-Exf[t][i][j][k+1]
                    +Byf[t][i][j][k+1]-(-nkExf[t-1][i][j][k+1]
                    +nkByf[t-1][i][j][k+1]+nkExf[t-1][i][j][k-1]
                    +nkByf[t-1][i][j][k-1])/2;
              nkEyf[t+1][i][j][k]=Eyf[t][i][j][k]-Bxf[t][i][j][k]+Eyf[t][i][j][k+1]
                    +Bxf[t][i][j][k+1]-(nkEyf[t-1][i][j][k+1]
                    +nkBxf[t-1][i][j][k+1]+nkEyf[t-1][i][j][k-1]
                    -nkBxf[t-1][i][j][k-1])/2;
              nkBxf[t+1][i][j][k]=(-Eyf[t][i][j][k])+Bxf[t][i][j][k]+Eyf[t][i][j][k+1]
                    +Bxf[t][i][j][k+1]-(nkEyf[t-1][i][j][k+1]
                    +nkBxf[t-1][i][j][k+1]-nkEyf[t-1][i][j][k-1]
                    +nkBxf[t-1][i][j][k-1])/2;
skip10:;
            }
         }
      }
      /* Calculate the node fields in the free space region   */
      for (i=1;i<=imaxf-1;++i)
      {   for (j=1;j<=jmaxf-1;++j)
          {   for (k=1;k<=kmaxf-1;++k)
              {
if (((k>=16)&&(k<=27))&&((i>=10)&&(i<=34))&&(j<=7)) goto skip11;
if (((k>=17)&&(k<=26))&&((i>=11)&&(i<=33))&&((j==8)||(j==9))) goto skip11;
      Exf[t+2][i][j][k]=Exf[t][i][j][k]+(njBzf[t+1][i][j][k]-njBzf[t+1][i][j-1][k]
                    -nkByf[t+1][i][j][k]+nkByf[t+1][i][j][k-1])/er;
      Eyf[t+2][i][j][k]=Eyf[t][i][j][k]+(nkBxf[t+1][i][j][k]-nkBxf[t+1][i][j][k-1]
                    -niBzf[t+1][i][j][k]+niBzf[t+1][i-1][j][k])/er;
      Ezf[t+2][i][j][k]=Ezf[t][i][j][k]+(niByf[t+1][i][j][k]-niByf[t+1][i-1][j][k]
                    -njBxf[t+1][i][j][k]+njBxf[t+1][i][j-1][k])/er;
      Bxf[t+2][i][j][k]=Bxf[t][i][j][k]+(nkEyf[t+1][i][j][k]-nkEyf[t+1][i][j][k-1]
                    -njEzf[t+1][i][j][k]+njEzf[t+1][i][j-1][k])/ur;
      Byf[t+2][i][j][k]=Byf[t][i][j][k]+(niEzf[t+1][i][j][k]-niEzf[t+1][i-1][j][k]
                    -nkExf[t+1][i][j][k]+nkExf[t+1][i][j][k-1])/ur;
      Bzf[t+2][i][j][k]=Bzf[t][i][j][k]+(njExf[t+1][i][j][k]-njExf[t+1][i][j-1][k]
                    -niEyf[t+1][i][j][k]+niEyf[t+1][i-1][j][k])/ur;
skip11:;
              }
          }
      }
       /*****************************************************************/
       /* Any output data should be taken here before the data exchange*/
```

```
/*****************************************************************/
/* Colour output data for Ez field at k=4 and k=20 TOP VIEW      */
/* Note: this calculation cannot deal with field values smaller */
/* than 4.5399929e-05  (absolute value) or greater than 1.0     */
/* Output colours are in a log scale .                          */
/*  for (j=1;j<=156;++j)  COMMENTED OUT !!!!!!!!!
   {   if (j<=113)
       {  for (i=1;i<=10;++i)
          {   a=0.0;
              fprintf(output1,"%d\n",(int)a);
          }
          for (i=1;i<=imax-1;++i)
          {   aa=fabs(log((double)fabs(Ez[t][i][j][4])));
              if (aa>10.0) aa=10.0;
              a=25.0*fabs(aa-10.0);
              if (a<2.0) a=2.0;
              fprintf(output1,"%d\n",(int)a);
          }
          for (i=1;i<=10;++i)
          {   a=0.0;
              fprintf(output1,"%d\n",(int)a);
          }
       }
       else
       {  if (j<=120)
          {  for (i=1;i<=10;++i)
             {   aa=fabs(log((double)fabs(Ezf[t][i][j-113][20])));
                 if (aa>10.0) aa=10.0;
                 a=25.0*fabs(aa-10.0);
                 if (a<2.0) a=2.0;
                 fprintf(output1,"%d\n",(int)a);
             }
             for (i=1;i<=imax-1;++i)
             {   aa=fabs(log((double)fabs(Ez[t][i][j][4])));
                 if (aa>10.0) aa=10.0;
                 a=25.0*fabs(aa-10.0);
                 if (a<2.0) a=2.0;
                 fprintf(output1,"%d\n",(int)a);
             }
             for (i=34;i<=imaxf-1;++i)
             {   aa=fabs(log((double)fabs(Ezf[t][i][j-113][20])));
                 if (aa>10.0) aa=10.0;
                 a=25.0*fabs(aa-10.0);
                 if (a<2.0) a=2.0;
                 fprintf(output1,"%d\n",(int)a);
             }
          }
          if ((j==121)||(j==122))
          {  for (i=1;i<=10;++i)
             {   aa=fabs(log((double)fabs(Ezf[t][i][j-113][20])));
                 if (aa>10.0) aa=10.0;
```

```
                a=25.0*fabs(aa-10.0);
                if (a<2.0) a=2;
                fprintf(output1,"%d\n",(int)a);
            }
            for (i=1;i<=imaxd-1;++i)
            {   aa=fabs(log((double)fabs(Ezd[t][i][j-120][4])));
                if (aa>10.0) aa=10.0;
                a=25.0*fabs(aa-10.0);
                if (a<2.0) a=2;
                fprintf(output1,"%d\n",(int)a);
            }
            for (i=34;i<=imaxf-1;++i)
            {   aa=fabs(log((double)fabs(Ezf[t][i][j-113][20])));
                if (aa>10.0) aa=10.0;
                a=25.0*fabs(aa-10.0);
                if (a<2.0) a=2;
                fprintf(output1,"%d\n",(int)a);
            }
          }
          if (j>=123)
          {  for (i=1;i<=imaxf-1;++i)
             {   aa=fabs(log((double)fabs(Ezf[t][i][j-113][20])));
                 if (aa>10.0) aa=10.0;
                 a=25.0*fabs(aa-10.0);
                 if (a<2.0) a=2;
                 fprintf(output1,"%d\n",(int)a);
             }
          }
       }
    }*/
/* Colour output data for Ez field at i=13 and i=24 SIDE VIEW    */
/* Note: this calculation cannot deal with field values smaller  */
/* than 4.5399929e-05  (absolute value) or greater than 1.0      */
/* Commented out for this iteration only.
    for (j=1;j<=156;++j)
    {   if (j<=113)
        {  for (k=1;k<=16;++k)
           {   a=0.0;
               fprintf(output2,"%d\n",(int)a);
           }
           for (k=1;k<=kmax-1;++k)
           {   aa=fabs(log((double)fabs(Ez[t][13][j][k])));
               if (aa>10.0) aa=10.0;
               a=25.0*fabs(aa-10.0);
               if (a<2.0) a=2.0;
               fprintf(output2,"%d\n",(int)a);
           }
           for (k=1;k<=17;++k)
           {   a=0.0;
               fprintf(output2,"%d\n",(int)a);
           }
```

```
}
else
{  if (j<=120)
   {  for (k=1;k<=16;++k)
      {   aa=fabs(log((double)fabs(Ezf[t][24][j-113][k])));
          if (aa>10.0) aa=10.0;
          a=25.0*fabs(aa-10.0);
          if (a<2.0) a=2.0;
          fprintf(output2,"%d\n",(int)a);
      }
      for (k=1;k<=kmax-1;++k)
      {   aa=fabs(log((double)fabs(Ez[t][13][j][k])));
          if (aa>10.0) aa=10.0;
          a=25.0*fabs(aa-10.0);
          if (a<2.0) a=2.0;
          fprintf(output2,"%d\n",(int)a);
      }
      for (k=27;k<=kmaxf-1;++k)
      {   aa=fabs(log((double)fabs(Ezf[t][24][j-113][k])));
          if (aa>10.0) aa=10.0;
          a=25.0*fabs(aa-10.0);
          if (a<2.0) a=2.0;
          fprintf(output2,"%d\n",(int)a);
      }
   }
   if ((j==121)||(j==122))
   {  for (k=1;k<=16;++k)
      {   aa=fabs(log((double)fabs(Ezf[t][24][j-113][k])));
          if (aa>10.0) aa=10.0;
          a=25.0*fabs(aa-10.0);
          if (a<2.0) a=2;
          fprintf(output2,"%d\n",(int)a);
      }
      for (k=1;k<=kmaxd-1;++k)
      {   aa=fabs(log((double)fabs(Ezd[t][13][j-120][k])));
          if (aa>10.0) aa=10.0;
          a=25.0*fabs(aa-10.0);
          if (a<2.0) a=2;
          fprintf(output2,"%d\n",(int)a);
      }
      for (k=27;k<=kmaxf-1;++k)
      {   aa=fabs(log((double)fabs(Ezf[t][24][j-113][k])));
          if (aa>10.0) aa=10.0;
          a=25.0*fabs(aa-10.0);
          if (a<2.0) a=2;
          fprintf(output2,"%d\n",(int)a);
      }
   }
   if (j>=123)
   {  for (k=1;k<=kmaxf-1;++k)
      {   aa=fabs(log((double)fabs(Ezf[t][24][j-113][k])));
```

```
                if (aa>10.0) aa=10.0;
                a=25.0*fabs(aa-10.0);
                if (a<2.0) a=2;
                fprintf(output2,"%d\n",(int)a);
             }
          }
       }
    }*/
 /*********************************************************************/
 /* This output data is for later FFT analysis. It is the TE10 mode*/
 /* contribution to the Ez field in the entire cross section of the*/
 /* guide.                                                            */
 /*********************************************************************/
for (j=20;j<=70;++j)
{   dumm1=0.0;
    dumm2=0.0;
    for (i=1;i<=imax-1;++i)
    {   for (k=1;k<=kmax-1;++k)
        {   dumm1=Ez[t][i][j][k]*sin(M_PI*(i-0.5)/(imax-1));
            dumm2=dumm2+dumm1;
        }
    }
    fprintf(output3,"%1.8e\n",2.0*dumm2/(23.0*10.0));
}
/*****************************************************************/
/* This section of code exchanges field components in time for */
/* the waveguide structure.                                      */
/*****************************************************************/

/* This exchange is for the absorbing boundary  */
for (k=1;k<=(kmax-1);++k)
{   for (i=1;i<=(imax-1);++i)
    {   for (j=0;j<=2;++j)
        {   njE1x[n-12][i][j][k]=njE1x[n-11][i][j][k];
            njE1z[n-12][i][j][k]=njE1z[n-11][i][j][k];
            njB1x[n-12][i][j][k]=njB1x[n-11][i][j][k];
            njB1z[n-12][i][j][k]=njB1z[n-11][i][j][k];

            njE1x[n-11][i][j][k]=njE1x[n-10][i][j][k];
            njE1z[n-11][i][j][k]=njE1z[n-10][i][j][k];
            njB1x[n-11][i][j][k]=njB1x[n-10][i][j][k];
            njB1z[n-11][i][j][k]=njB1z[n-10][i][j][k];

            njE1x[n-10][i][j][k]=njE1x[n-9][i][j][k];
            njE1z[n-10][i][j][k]=njE1z[n-9][i][j][k];
            njB1x[n-10][i][j][k]=njB1x[n-9][i][j][k];
            njB1z[n-10][i][j][k]=njB1z[n-9][i][j][k];

            njE1x[n-9][i][j][k]=njE1x[n-8][i][j][k];
            njE1z[n-9][i][j][k]=njE1z[n-8][i][j][k];
            njB1x[n-9][i][j][k]=njB1x[n-8][i][j][k];
            njB1z[n-9][i][j][k]=njB1z[n-8][i][j][k];
```

```
            njE1x[n-8][i][j][k]=njE1x[n-7][i][j][k];
            njE1z[n-8][i][j][k]=njE1z[n-7][i][j][k];
            njB1x[n-8][i][j][k]=njB1x[n-7][i][j][k];
            njB1z[n-8][i][j][k]=njB1z[n-7][i][j][k];

            njE1x[n-7][i][j][k]=njE1x[n-6][i][j][k];
            njE1z[n-7][i][j][k]=njE1z[n-6][i][j][k];
            njB1x[n-7][i][j][k]=njB1x[n-6][i][j][k];
            njB1z[n-7][i][j][k]=njB1z[n-6][i][j][k];

            njE1x[n-6][i][j][k]=njE1x[n-5][i][j][k];
            njE1z[n-6][i][j][k]=njE1z[n-5][i][j][k];
            njB1x[n-6][i][j][k]=njB1x[n-5][i][j][k];
            njB1z[n-6][i][j][k]=njB1z[n-5][i][j][k];

            njE1x[n-5][i][j][k]=njE1x[n-4][i][j][k];
            njE1z[n-5][i][j][k]=njE1z[n-4][i][j][k];
            njB1x[n-5][i][j][k]=njB1x[n-4][i][j][k];
            njB1z[n-5][i][j][k]=njB1z[n-4][i][j][k];

            njE1x[n-4][i][j][k]=njE1x[n-3][i][j][k];
            njE1z[n-4][i][j][k]=njE1z[n-3][i][j][k];
            njB1x[n-4][i][j][k]=njB1x[n-3][i][j][k];
            njB1z[n-4][i][j][k]=njB1z[n-3][i][j][k];

            njE1x[n-3][i][j][k]=njE1x[n-2][i][j][k];
            njE1z[n-3][i][j][k]=njE1z[n-2][i][j][k];
            njB1x[n-3][i][j][k]=njB1x[n-2][i][j][k];
            njB1z[n-3][i][j][k]=njB1z[n-2][i][j][k];

            njE1x[n-2][i][j][k]=njEx[t-1][i][2*j+2][k];
            njE1z[n-2][i][j][k]=njEz[t-1][i][2*j+2][k];
            njB1x[n-2][i][j][k]=njBx[t-1][i][2*j+2][k];
            njB1z[n-2][i][j][k]=njBz[t-1][i][2*j+2][k];
        }
    }
}
/*  This exchange is for the interior of the guide  */
for (k=0;k<=(kmax-1);(k=k+1))
{   for (j=0;j<=(jmax-1);(j=j+1))
    {   for (i=0;i<=(imax-1);(i=i+1))
        {   njEx[t-1][i][j][k]=njEx[t+1][i][j][k];
            nkEx[t-1][i][j][k]=nkEx[t+1][i][j][k];
            njBx[t-1][i][j][k]=njBx[t+1][i][j][k];
            nkBx[t-1][i][j][k]=nkBx[t+1][i][j][k];
            niEy[t-1][i][j][k]=niEy[t+1][i][j][k];
            nkEy[t-1][i][j][k]=nkEy[t+1][i][j][k];
            niBy[t-1][i][j][k]=niBy[t+1][i][j][k];
            nkBy[t-1][i][j][k]=nkBy[t+1][i][j][k];
            niEz[t-1][i][j][k]=niEz[t+1][i][j][k];
            njEz[t-1][i][j][k]=njEz[t+1][i][j][k];
```

```
            niBz[t-1][i][j][k]=niBz[t+1][i][j][k];
            njBz[t-1][i][j][k]=njBz[t+1][i][j][k];

            Ex[t][i][j][k]=Ex[t+2][i][j][k];
            Ey[t][i][j][k]=Ey[t+2][i][j][k];
            Ez[t][i][j][k]=Ez[t+2][i][j][k];
            Bx[t][i][j][k]=Bx[t+2][i][j][k];
            By[t][i][j][k]=By[t+2][i][j][k];
            Bz[t][i][j][k]=Bz[t+2][i][j][k];
         }
      }
   }
/*****************************************************************/
/* This section of code exchanges field components in time for */
/* the dielectric structure.                                     */
/*****************************************************************/
   /*  This exchange is for the interior of the dielectric  */
   for (k=0;k<=(kmaxd-1);(k=k+1))
   {  for (j=0;j<=(jmaxd-1);(j=j+1))
      {  for (i=0;i<=(imaxd-1);(i=i+1))
         {  njExd[t-1][i][j][k]=njExd[t+1][i][j][k];
            nkExd[t-1][i][j][k]=nkExd[t+1][i][j][k];
            njBxd[t-1][i][j][k]=njBxd[t+1][i][j][k];
            nkBxd[t-1][i][j][k]=nkBxd[t+1][i][j][k];
            niEyd[t-1][i][j][k]=niEyd[t+1][i][j][k];
            nkEyd[t-1][i][j][k]=nkEyd[t+1][i][j][k];
            niByd[t-1][i][j][k]=niByd[t+1][i][j][k];
            nkByd[t-1][i][j][k]=nkByd[t+1][i][j][k];
            niEzd[t-1][i][j][k]=niEzd[t+1][i][j][k];
            njEzd[t-1][i][j][k]=njEzd[t+1][i][j][k];
            niBzd[t-1][i][j][k]=niBzd[t+1][i][j][k];
            njBzd[t-1][i][j][k]=njBzd[t+1][i][j][k];

            Exd[t][i][j][k]=Exd[t+2][i][j][k];
            Eyd[t][i][j][k]=Eyd[t+2][i][j][k];
            Ezd[t][i][j][k]=Ezd[t+2][i][j][k];
            Bxd[t][i][j][k]=Bxd[t+2][i][j][k];
            Byd[t][i][j][k]=Byd[t+2][i][j][k];
            Bzd[t][i][j][k]=Bzd[t+2][i][j][k];
         }
      }
   }
/*****************************************************************/
/* This section of code exchanges field components in time for */
/* the free space region.                                        */
/*****************************************************************/
/* This exchange is for the absorbing walls to the left and right  */
   for (k=1;k<=(kmaxf-1);++k)
   {  for (i=1;i<=(imaxf-1);++i)
      {  for (j=0;j<=2;++j)
         {  if (((k>=16)&&(k<=27))&&((i>=10)&&(i<=34))) goto skip12;
```

```
njE1xf[n-12][i][j][k]=njE1xf[n-11][i][j][k];
njE1zf[n-12][i][j][k]=njE1zf[n-11][i][j][k];
njB1xf[n-12][i][j][k]=njB1xf[n-11][i][j][k];
njB1zf[n-12][i][j][k]=njB1zf[n-11][i][j][k];

njE1xf[n-11][i][j][k]=njE1xf[n-10][i][j][k];
njE1zf[n-11][i][j][k]=njE1zf[n-10][i][j][k];
njB1xf[n-11][i][j][k]=njB1xf[n-10][i][j][k];
njB1zf[n-11][i][j][k]=njB1zf[n-10][i][j][k];

njE1xf[n-10][i][j][k]=njE1xf[n-9][i][j][k];
njE1zf[n-10][i][j][k]=njE1zf[n-9][i][j][k];
njB1xf[n-10][i][j][k]=njB1xf[n-9][i][j][k];
njB1zf[n-10][i][j][k]=njB1zf[n-9][i][j][k];

njE1xf[n-9][i][j][k]=njE1xf[n-8][i][j][k];
njE1zf[n-9][i][j][k]=njE1zf[n-8][i][j][k];
njB1xf[n-9][i][j][k]=njB1xf[n-8][i][j][k];
njB1zf[n-9][i][j][k]=njB1zf[n-8][i][j][k];

njE1xf[n-8][i][j][k]=njE1xf[n-7][i][j][k];
njE1zf[n-8][i][j][k]=njE1zf[n-7][i][j][k];
njB1xf[n-8][i][j][k]=njB1xf[n-7][i][j][k];
njB1zf[n-8][i][j][k]=njB1zf[n-7][i][j][k];

njE1xf[n-7][i][j][k]=njE1xf[n-6][i][j][k];
njE1zf[n-7][i][j][k]=njE1zf[n-6][i][j][k];
njB1xf[n-7][i][j][k]=njB1xf[n-6][i][j][k];
njB1zf[n-7][i][j][k]=njB1zf[n-6][i][j][k];

njE1xf[n-6][i][j][k]=njE1xf[n-5][i][j][k];
njE1zf[n-6][i][j][k]=njE1zf[n-5][i][j][k];
njB1xf[n-6][i][j][k]=njB1xf[n-5][i][j][k];
njB1zf[n-6][i][j][k]=njB1zf[n-5][i][j][k];

njE1xf[n-5][i][j][k]=njE1xf[n-4][i][j][k];
njE1zf[n-5][i][j][k]=njE1zf[n-4][i][j][k];
njB1xf[n-5][i][j][k]=njB1xf[n-4][i][j][k];
njB1zf[n-5][i][j][k]=njB1zf[n-4][i][j][k];

njE1xf[n-4][i][j][k]=njE1xf[n-3][i][j][k];
njE1zf[n-4][i][j][k]=njE1zf[n-3][i][j][k];
njB1xf[n-4][i][j][k]=njB1xf[n-3][i][j][k];
njB1zf[n-4][i][j][k]=njB1zf[n-3][i][j][k];
njE1xf[n-3][i][j][k]=njE1xf[n-2][i][j][k];
njE1zf[n-3][i][j][k]=njE1zf[n-2][i][j][k];
njB1xf[n-3][i][j][k]=njB1xf[n-2][i][j][k];
njB1zf[n-3][i][j][k]=njB1zf[n-2][i][j][k];

njE1xf[n-2][i][j][k]=njExf[t-1][i][2*j+2][k];
njE1zf[n-2][i][j][k]=njEzf[t-1][i][2*j+2][k];
```

```
    njB1xf[n-2][i][j][k]=njBxf[t-1][i][2*j+2][k];
    njB1zf[n-2][i][j][k]=njBzf[t-1][i][2*j+2][k];
    skip12:;
}
for (j=0;j<=2;++j)
{   njE2xf[n-12][i][j][k]=njE2xf[n-11][i][j][k];
    njE2zf[n-12][i][j][k]=njE2zf[n-11][i][j][k];
    njB2xf[n-12][i][j][k]=njB2xf[n-11][i][j][k];
    njB2zf[n-12][i][j][k]=njB2zf[n-11][i][j][k];

    njE2xf[n-11][i][j][k]=njE2xf[n-10][i][j][k];
    njE2zf[n-11][i][j][k]=njE2zf[n-10][i][j][k];
    njB2xf[n-11][i][j][k]=njB2xf[n-10][i][j][k];
    njB2zf[n-11][i][j][k]=njB2zf[n-10][i][j][k];

    njE2xf[n-10][i][j][k]=njE2xf[n-9][i][j][k];
    njE2zf[n-10][i][j][k]=njE2zf[n-9][i][j][k];
    njB2xf[n-10][i][j][k]=njB2xf[n-9][i][j][k];
    njB2zf[n-10][i][j][k]=njB2zf[n-9][i][j][k];
    njE2xf[n-9][i][j][k]=njE2xf[n-8][i][j][k];
    njE2zf[n-9][i][j][k]=njE2zf[n-8][i][j][k];
    njB2xf[n-9][i][j][k]=njB2xf[n-8][i][j][k];
    njB2zf[n-9][i][j][k]=njB2zf[n-8][i][j][k];
    njE2xf[n-8][i][j][k]=njE2xf[n-7][i][j][k];
    njE2zf[n-8][i][j][k]=njE2zf[n-7][i][j][k];
    njB2xf[n-8][i][j][k]=njB2xf[n-7][i][j][k];
    njB2zf[n-8][i][j][k]=njB2zf[n-7][i][j][k];

    njE2xf[n-7][i][j][k]=njE2xf[n-6][i][j][k];
    njE2zf[n-7][i][j][k]=njE2zf[n-6][i][j][k];
    njB2xf[n-7][i][j][k]=njB2xf[n-6][i][j][k];
    njB2zf[n-7][i][j][k]=njB2zf[n-6][i][j][k];

    njE2xf[n-6][i][j][k]=njE2xf[n-5][i][j][k];
    njE2zf[n-6][i][j][k]=njE2zf[n-5][i][j][k];
    njB2xf[n-6][i][j][k]=njB2xf[n-5][i][j][k];
    njB2zf[n-6][i][j][k]=njB2zf[n-5][i][j][k];

    njE2xf[n-5][i][j][k]=njE2xf[n-4][i][j][k];
    njE2zf[n-5][i][j][k]=njE2zf[n-4][i][j][k];
    njB2xf[n-5][i][j][k]=njB2xf[n-4][i][j][k];
    njB2zf[n-5][i][j][k]=njB2zf[n-4][i][j][k];

    njE2xf[n-4][i][j][k]=njE2xf[n-3][i][j][k];
    njE2zf[n-4][i][j][k]=njE2zf[n-3][i][j][k];
    njB2xf[n-4][i][j][k]=njB2xf[n-3][i][j][k];
    njB2zf[n-4][i][j][k]=njB2zf[n-3][i][j][k];

    njE2xf[n-3][i][j][k]=njE2xf[n-2][i][j][k];
    njE2zf[n-3][i][j][k]=njE2zf[n-2][i][j][k];
    njB2xf[n-3][i][j][k]=njB2xf[n-2][i][j][k];
```

```
            njB2zf[n-3][i][j][k]=njB2zf[n-2][i][j][k];
            njE2xf[n-2][i][j][k]=njExf[t-1][i][jmaxf-7+2*j][k];
            njE2zf[n-2][i][j][k]=njEzf[t-1][i][jmaxf-7+2*j][k];
            njB2xf[n-2][i][j][k]=njBxf[t-1][i][jmaxf-7+2*j][k];
            njB2zf[n-2][i][j][k]=njBzf[t-1][i][jmaxf-7+2*j][k];
        }
    }
}
/* Exchange for the absorbing walls on top and on the bottom   */
for (j=1;j<=(jmaxf-1);++j)
{   for (i=1;i<=(imaxf-1);++i)
    {   for (k=0;k<=2;++k)
        {   nkE1xf[n-12][i][j][k]=nkE1xf[n-11][i][j][k];
            nkE1yf[n-12][i][j][k]=nkE1yf[n-11][i][j][k];
            nkB1xf[n-12][i][j][k]=nkB1xf[n-11][i][j][k];
            nkB1yf[n-12][i][j][k]=nkB1yf[n-11][i][j][k];
            nkE1xf[n-11][i][j][k]=nkE1xf[n-10][i][j][k];
            nkE1yf[n-11][i][j][k]=nkE1yf[n-10][i][j][k];
            nkB1xf[n-11][i][j][k]=nkB1xf[n-10][i][j][k];
            nkB1yf[n-11][i][j][k]=nkB1yf[n-10][i][j][k];
            nkE1xf[n-10][i][j][k]=nkE1xf[n-9][i][j][k];
            nkE1yf[n-10][i][j][k]=nkE1yf[n-9][i][j][k];
            nkB1xf[n-10][i][j][k]=nkB1xf[n-9][i][j][k];
            nkB1yf[n-10][i][j][k]=nkB1yf[n-9][i][j][k];

            nkE1xf[n-9][i][j][k]=nkE1xf[n-8][i][j][k];
            nkE1yf[n-9][i][j][k]=nkE1yf[n-8][i][j][k];
            nkB1xf[n-9][i][j][k]=nkB1xf[n-8][i][j][k];
            nkB1yf[n-9][i][j][k]=nkB1yf[n-8][i][j][k];

            nkE1xf[n-8][i][j][k]=nkE1xf[n-7][i][j][k];
            nkE1yf[n-8][i][j][k]=nkE1yf[n-7][i][j][k];
            nkB1xf[n-8][i][j][k]=nkB1xf[n-7][i][j][k];
            nkB1yf[n-8][i][j][k]=nkB1yf[n-7][i][j][k];

            nkE1xf[n-7][i][j][k]=nkE1xf[n-6][i][j][k];
            nkE1yf[n-7][i][j][k]=nkE1yf[n-6][i][j][k];
            nkB1xf[n-7][i][j][k]=nkB1xf[n-6][i][j][k];
            nkB1yf[n-7][i][j][k]=nkB1yf[n-6][i][j][k];

            nkE1xf[n-6][i][j][k]=nkE1xf[n-5][i][j][k];
            nkE1yf[n-6][i][j][k]=nkE1yf[n-5][i][j][k];
            nkB1xf[n-6][i][j][k]=nkB1xf[n-5][i][j][k];
            nkB1yf[n-6][i][j][k]=nkB1yf[n-5][i][j][k];

            nkE1xf[n-5][i][j][k]=nkE1xf[n-4][i][j][k];
            nkE1yf[n-5][i][j][k]=nkE1yf[n-4][i][j][k];
            nkB1xf[n-5][i][j][k]=nkB1xf[n-4][i][j][k];
            nkB1yf[n-5][i][j][k]=nkB1yf[n-4][i][j][k];
            nkE1xf[n-4][i][j][k]=nkE1xf[n-3][i][j][k];
            nkE1yf[n-4][i][j][k]=nkE1yf[n-3][i][j][k];
```

```
    nkB1xf[n-4][i][j][k]=nkB1xf[n-3][i][j][k];
    nkB1yf[n-4][i][j][k]=nkB1yf[n-3][i][j][k];

    nkE1xf[n-3][i][j][k]=nkE1xf[n-2][i][j][k];
    nkE1yf[n-3][i][j][k]=nkE1yf[n-2][i][j][k];
    nkB1xf[n-3][i][j][k]=nkB1xf[n-2][i][j][k];
    nkB1yf[n-3][i][j][k]=nkB1yf[n-2][i][j][k];

    nkE1xf[n-2][i][j][k]=nkExf[t-1][i][j][2*k+2];
    nkE1yf[n-2][i][j][k]=nkEyf[t-1][i][j][2*k+2];
    nkB1xf[n-2][i][j][k]=nkBxf[t-1][i][j][2*k+2];
    nkB1yf[n-2][i][j][k]=nkByf[t-1][i][j][2*k+2];
}
for (k=0;k<=2;++k)
{   nkE2xf[n-12][i][j][k]=nkE2xf[n-11][i][j][k];
    nkE2yf[n-12][i][j][k]=nkE2yf[n-11][i][j][k];
    nkB2xf[n-12][i][j][k]=nkB2xf[n-11][i][j][k];
    nkB2yf[n-12][i][j][k]=nkB2yf[n-11][i][j][k];

    nkE2xf[n-11][i][j][k]=nkE2xf[n-10][i][j][k];
    nkE2yf[n-11][i][j][k]=nkE2yf[n-10][i][j][k];
    nkB2xf[n-11][i][j][k]=nkB2xf[n-10][i][j][k];
    nkB2yf[n-11][i][j][k]=nkB2yf[n-10][i][j][k];

    nkE2xf[n-10][i][j][k]=nkE2xf[n-9][i][j][k];
    nkE2yf[n-10][i][j][k]=nkE2yf[n-9][i][j][k];
    nkB2xf[n-10][i][j][k]=nkB2xf[n-9][i][j][k];
    nkB2yf[n-10][i][j][k]=nkB2yf[n-9][i][j][k];

    nkE2xf[n-9][i][j][k]=nkE2xf[n-8][i][j][k];
    nkE2yf[n-9][i][j][k]=nkE2yf[n-8][i][j][k];
    nkB2xf[n-9][i][j][k]=nkB2xf[n-8][i][j][k];
    nkB2yf[n-9][i][j][k]=nkB2yf[n-8][i][j][k];

    nkE2xf[n-8][i][j][k]=nkE2xf[n-7][i][j][k];
    nkE2yf[n-8][i][j][k]=nkE2yf[n-7][i][j][k];
    nkB2xf[n-8][i][j][k]=nkB2xf[n-7][i][j][k];
    nkB2yf[n-8][i][j][k]=nkB2yf[n-7][i][j][k];

    nkE2xf[n-7][i][j][k]=nkE2xf[n-6][i][j][k];
    nkE2yf[n-7][i][j][k]=nkE2yf[n-6][i][j][k];
    nkB2xf[n-7][i][j][k]=nkB2xf[n-6][i][j][k];
    nkB2yf[n-7][i][j][k]=nkB2yf[n-6][i][j][k];

    nkE2xf[n-6][i][j][k]=nkE2xf[n-5][i][j][k];
    nkE2yf[n-6][i][j][k]=nkE2yf[n-5][i][j][k];
    nkB2xf[n-6][i][j][k]=nkB2xf[n-5][i][j][k];
    nkB2yf[n-6][i][j][k]=nkB2yf[n-5][i][j][k];

    nkE2xf[n-5][i][j][k]=nkE2xf[n-4][i][j][k];
    nkE2yf[n-5][i][j][k]=nkE2yf[n-4][i][j][k];
```

```
            nkB2xf[n-5][i][j][k]=nkB2xf[n-4][i][j][k];
            nkB2yf[n-5][i][j][k]=nkB2yf[n-4][i][j][k];

            nkE2xf[n-4][i][j][k]=nkE2xf[n-3][i][j][k];
            nkE2yf[n-4][i][j][k]=nkE2yf[n-3][i][j][k];
            nkB2xf[n-4][i][j][k]=nkB2xf[n-3][i][j][k];
            nkB2yf[n-4][i][j][k]=nkB2yf[n-3][i][j][k];
            nkE2xf[n-3][i][j][k]=nkE2xf[n-2][i][j][k];
            nkE2yf[n-3][i][j][k]=nkE2yf[n-2][i][j][k];
            nkB2xf[n-3][i][j][k]=nkB2xf[n-2][i][j][k];
            nkB2yf[n-3][i][j][k]=nkB2yf[n-2][i][j][k];

            nkE2xf[n-2][i][j][k]=nkExf[t-1][i][j][kmaxf-7+2*k];
            nkE2yf[n-2][i][j][k]=nkEyf[t-1][i][j][kmaxf-7+2*k];
            nkB2xf[n-2][i][j][k]=nkBxf[t-1][i][j][kmaxf-7+2*k];
            nkB2yf[n-2][i][j][k]=nkByf[t-1][i][j][kmaxf-7+2*k];
        }
    }
}
/* This exchange is for the absorbing walls in front and in the back   */
for (j=1;j<=(jmaxf-1);++j)
{   for (k=1;k<=(kmaxf-1);++k)
    {   for (i=0;i<=2;++i)
        {   niE1zf[n-12][i][j][k]=niE1zf[n-11][i][j][k];
            niE1yf[n-12][i][j][k]=niE1yf[n-11][i][j][k];
            niB1zf[n-12][i][j][k]=niB1zf[n-11][i][j][k];
            niB1yf[n-12][i][j][k]=niB1yf[n-11][i][j][k];

            niE1zf[n-11][i][j][k]=niE1zf[n-10][i][j][k];
            niE1yf[n-11][i][j][k]=niE1yf[n-10][i][j][k];
            niB1zf[n-11][i][j][k]=niB1zf[n-10][i][j][k];
            niB1yf[n-11][i][j][k]=niB1yf[n-10][i][j][k];

            niE1zf[n-10][i][j][k]=niE1zf[n-9][i][j][k];
            niE1yf[n-10][i][j][k]=niE1yf[n-9][i][j][k];
            niB1zf[n-10][i][j][k]=niB1zf[n-9][i][j][k];
            niB1yf[n-10][i][j][k]=niB1yf[n-9][i][j][k];
            niE1zf[n-9][i][j][k]=niE1zf[n-8][i][j][k];
            niE1yf[n-9][i][j][k]=niE1yf[n-8][i][j][k];
            niB1zf[n-9][i][j][k]=niB1zf[n-8][i][j][k];
            niB1yf[n-9][i][j][k]=niB1yf[n-8][i][j][k];

            niE1zf[n-8][i][j][k]=niE1zf[n-7][i][j][k];
            niE1yf[n-8][i][j][k]=niE1yf[n-7][i][j][k];
            niB1zf[n-8][i][j][k]=niB1zf[n-7][i][j][k];
            niB1yf[n-8][i][j][k]=niB1yf[n-7][i][j][k];

            niE1zf[n-7][i][j][k]=niE1zf[n-6][i][j][k];
            niE1yf[n-7][i][j][k]=niE1yf[n-6][i][j][k];
            niB1zf[n-7][i][j][k]=niB1zf[n-6][i][j][k];
            niB1yf[n-7][i][j][k]=niB1yf[n-6][i][j][k];
```

```
    niE1zf[n-6][i][j][k]=niE1zf[n-5][i][j][k];
    niE1yf[n-6][i][j][k]=niE1yf[n-5][i][j][k];
    niB1zf[n-6][i][j][k]=niB1zf[n-5][i][j][k];
    niB1yf[n-6][i][j][k]=niB1yf[n-5][i][j][k];
    niE1zf[n-5][i][j][k]=niE1zf[n-4][i][j][k];
    niE1yf[n-5][i][j][k]=niE1yf[n-4][i][j][k];
    niB1zf[n-5][i][j][k]=niB1zf[n-4][i][j][k];
    niB1yf[n-5][i][j][k]=niB1yf[n-4][i][j][k];

    niE1zf[n-4][i][j][k]=niE1zf[n-3][i][j][k];
    niE1yf[n-4][i][j][k]=niE1yf[n-3][i][j][k];
    niB1zf[n-4][i][j][k]=niB1zf[n-3][i][j][k];
    niB1yf[n-4][i][j][k]=niB1yf[n-3][i][j][k];

    niE1zf[n-3][i][j][k]=niE1zf[n-2][i][j][k];
    niE1yf[n-3][i][j][k]=niE1yf[n-2][i][j][k];
    niB1zf[n-3][i][j][k]=niB1zf[n-2][i][j][k];
    niB1yf[n-3][i][j][k]=niB1yf[n-2][i][j][k];
    niE1zf[n-2][i][j][k]=niEzf[t-1][2*i+2][j][k];
    niE1yf[n-2][i][j][k]=niEyf[t-1][2*i+2][j][k];
    niB1zf[n-2][i][j][k]=niBzf[t-1][2*i+2][j][k];
    niB1yf[n-2][i][j][k]=niByf[t-1][2*i+2][j][k];
}
for (i=0;i<=2;++i)
{   niE2zf[n-12][i][j][k]=niE2zf[n-11][i][j][k];
    niE2yf[n-12][i][j][k]=niE2yf[n-11][i][j][k];
    niB2zf[n-12][i][j][k]=niB2zf[n-11][i][j][k];
    niB2yf[n-12][i][j][k]=niB2yf[n-11][i][j][k];

    niE2zf[n-11][i][j][k]=niE2zf[n-10][i][j][k];
    niE2yf[n-11][i][j][k]=niE2yf[n-10][i][j][k];
    niB2zf[n-11][i][j][k]=niB2zf[n-10][i][j][k];
    niB2yf[n-11][i][j][k]=niB2yf[n-10][i][j][k];

    niE2zf[n-10][i][j][k]=niE2zf[n-9][i][j][k];
    niE2yf[n-10][i][j][k]=niE2yf[n-9][i][j][k];
    niB2zf[n-10][i][j][k]=niB2zf[n-9][i][j][k];
    niB2yf[n-10][i][j][k]=niB2yf[n-9][i][j][k];

    niE2zf[n-9][i][j][k]=niE2zf[n-8][i][j][k];
    niE2yf[n-9][i][j][k]=niE2yf[n-8][i][j][k];
    niB2zf[n-9][i][j][k]=niB2zf[n-8][i][j][k];
    niB2yf[n-9][i][j][k]=niB2yf[n-8][i][j][k];

    niE2zf[n-8][i][j][k]=niE2zf[n-7][i][j][k];
    niE2yf[n-8][i][j][k]=niE2yf[n-7][i][j][k];
    niB2zf[n-8][i][j][k]=niB2zf[n-7][i][j][k];
    niB2yf[n-8][i][j][k]=niB2yf[n-7][i][j][k];

    niE2zf[n-7][i][j][k]=niE2zf[n-6][i][j][k];
    niE2yf[n-7][i][j][k]=niE2yf[n-6][i][j][k];
```

```
                niB2zf[n-7][i][j][k]=niB2zf[n-6][i][j][k];
                niB2yf[n-7][i][j][k]=niB2yf[n-6][i][j][k];
                niE2zf[n-6][i][j][k]=niE2zf[n-5][i][j][k];
                niE2yf[n-6][i][j][k]=niE2yf[n-5][i][j][k];
                niB2zf[n-6][i][j][k]=niB2zf[n-5][i][j][k];
                niB2yf[n-6][i][j][k]=niB2yf[n-5][i][j][k];

                niE2zf[n-5][i][j][k]=niE2zf[n-4][i][j][k];
                niE2yf[n-5][i][j][k]=niE2yf[n-4][i][j][k];
                niB2zf[n-5][i][j][k]=niB2zf[n-4][i][j][k];
                niB2yf[n-5][i][j][k]=niB2yf[n-4][i][j][k];

                niE2zf[n-4][i][j][k]=niE2zf[n-3][i][j][k];
                niE2yf[n-4][i][j][k]=niE2yf[n-3][i][j][k];
                niB2zf[n-4][i][j][k]=niB2zf[n-3][i][j][k];
                niB2yf[n-4][i][j][k]=niB2yf[n-3][i][j][k];

                niE2zf[n-3][i][j][k]=niE2zf[n-2][i][j][k];
                niE2yf[n-3][i][j][k]=niE2yf[n-2][i][j][k];
                niB2zf[n-3][i][j][k]=niB2zf[n-2][i][j][k];
                niB2yf[n-3][i][j][k]=niB2yf[n-2][i][j][k];

                niE2zf[n-2][i][j][k]=niEzf[t-1][imaxf-7+2*i][j][k];
                niE2yf[n-2][i][j][k]=niEyf[t-1][imaxf-7+2*i][j][k];
                niB2zf[n-2][i][j][k]=niBzf[t-1][imaxf-7+2*i][j][k];
                niB2yf[n-2][i][j][k]=niByf[t-1][imaxf-7+2*i][j][k];
            }
        }
    }
    /*  This exchange is for all field components in the free space region */
    for (k=0;k<=(kmaxf-1);(k=k+1))
    {   for (j=0;j<=(jmaxf-1);(j=j+1))
        {   for (i=0;i<=(imaxf-1);(i=i+1))
            {
if (((k>=16)&&(k<=27))&&((i>=10)&&(i<=34))&&(j<=7)) goto skip14;
if (((k>=17)&&(k<=26))&&((i>=11)&&(i<=33))&&((j==8)||(j==9))) goto skip14;
                Exf[t][i][j][k]=Exf[t+2][i][j][k];
                Eyf[t][i][j][k]=Eyf[t+2][i][j][k];
                Ezf[t][i][j][k]=Ezf[t+2][i][j][k];
                Bxf[t][i][j][k]=Bxf[t+2][i][j][k];
                Byf[t][i][j][k]=Byf[t+2][i][j][k];
                Bzf[t][i][j][k]=Bzf[t+2][i][j][k];
skip14:;
if (((k>=16)&&(k<=26))&&((i>=10)&&(i<=33))&&(j<=6)) goto skip15;
if (((k>=17)&&(k<=25))&&((i>=11)&&(i<=32))&&(j==8)) goto skip15;
                njExf[t-1][i][j][k]=njExf[t+1][i][j][k];
                nkExf[t-1][i][j][k]=nkExf[t+1][i][j][k];
                njBxf[t-1][i][j][k]=njBxf[t+1][i][j][k];
                nkBxf[t-1][i][j][k]=nkBxf[t+1][i][j][k];
                niEyf[t-1][i][j][k]=niEyf[t+1][i][j][k];
                nkEyf[t-1][i][j][k]=nkEyf[t+1][i][j][k];
```

```
                niByf[t-1][i][j][k]=niByf[t+1][i][j][k];
                nkByf[t-1][i][j][k]=nkByf[t+1][i][j][k];
                niEzf[t-1][i][j][k]=niEzf[t+1][i][j][k];
                njEzf[t-1][i][j][k]=njEzf[t+1][i][j][k];
                niBzf[t-1][i][j][k]=niBzf[t+1][i][j][k];
                njBzf[t-1][i][j][k]=njBzf[t+1][i][j][k];
skip15:;
            }
        }
    }
}    /* Now step to the next iteration in time  */
printf("All calculations completed.\n");
fclose(output1);
fclose(output2);
fclose(output3);
}   /* Close main.   */
```

Appendix C

Computer Program To Determine the Standing-Wave Pattern

This program calculates the standing-wave pattern over a specified range of frequencies from data contained in a file called exparfft.dat. It assumes that field data for the TE_{10} mode (E_z field component in this case) for 50 waveguide cross sections have been stored in the exparfft.dat file for 1000 time iterations. The FFT algorithm is the computer coded version of equation 3.5 of chapter 3.3. The output file, raw.dat, contains the raw data of the standing-wave pattern for various frequencies.

```
#include <stdio.h>
#include <math.h>
FILE *input1,*fopen();
FILE *output1,*fopen();
float FFT();
main()
{
static float data[1001][51];
static float Ez[51][31];
float freq,delta,freqinc;
float c;
int m,j,step,NSTEP,jmax,freqnum;
input1=fopen("exparfft.dat","r");
output1=fopen("raw.dat","w");

jmax=50;
NSTEP=1000;
delta=0.001;
freqinc=0.2;    /*freq increment in GHz */
freqnum=21;     /* number of frequency data points  */
c=3.0e8;

printf("Now processing FFT data for FD-TD simulation from file exparfft.dat.\n");
```

```
printf("(%d iterations)\n",NSTEP);

/*read in data from file exparfft.dat*/
for (step=1;step<=NSTEP;++step)
{   for (j=0;j<=jmax;++j)
    {   fscanf(input1,"%e",&data[step][j]);
    }
}

freq=4.0;  /*start freq. in GHz is twice value shown here  */
/* data below is corrected for er=ur=1 */
for (m=1;m<=freqnum;++m)
{   for (j=0;j<=jmax;++j)
    {   Ez[j][m]=FFT(j,data,freq,NSTEP,delta);
        fprintf(output1,"freq(GHZ)=%1.3f\tEz[%d][%d]=%1.8f\n"
         ,2.0*freq,j,m,1.4142*Ez[j][m]);
    }
    printf("Doing FFT for freq of %1.3f GHz\n",2.0*freq);
    freq=freq+(freqinc/2.0);
}

printf("All calculations complete, raw data written to raw.dat\n");
fclose(output1);
fclose(input1);
}  /*end main*/

/*********************************************************/
/* The function FFT calculates the fast fourier          */
/* transform of a given data array ddata for a part-     */
/* icular frequency ffreq (in GHz), at jj for a ddelta   */
/* and total number of iterations NNSTEP.                */
/*********************************************************/
float FFT(jj,ddata,ffreq,NNSTEP,ddelta)
float ddata[1001][51];
float ffreq,ddelta;
int jj,NNSTEP;
{
float dum1,dum2,dum3,dum4,mag;
int sstep;
dum2=0.0;
dum4=0.0;
for (sstep=1;sstep<=NNSTEP;++sstep)
{
    dum1=ddata[sstep][jj]*(cos((double)(2*M_PI*sstep*ddelta*3.333333* ffreq)));
    dum3=ddata[sstep][jj]*(sin((double)(2*M_PI*sstep*ddelta*3.333333* ffreq)));
    dum2=dum2+dum1;
    dum4=dum4+dum3;
}
mag=sqrt(dum2*dum2+dum4*dum4);
return(mag);
}
```