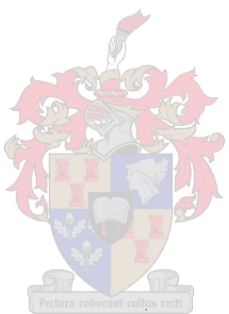
Electromagnetic Modelling of a Borehole Radar Environment with the Finite Difference Time Domain Method

Ernst H. Burger



Thesis presented in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Stellenbosch.

Advisor

Prof. David B. Davidson

December 2000

– Declaration –

"I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree."

Signature ..... Ernst H. Burger

Date .....20-11-2000.....

Abstract

South Africa has an immensely rich reserve of minerals which still has to be exploited. The problem with these reserves is that they exist in reefs where the mining environment is extremely hazardous, and where mining is very expensive. These are only two of the reasons why borehole radar has recently become a very important field of research in the South African mining industry. These radars have to operate in rock, which has a number of electromagnetically problematic characteristics, which greatly complicate modelling and design of suitable radars. The goal of this project is to demonstrate how the Finite Difference Time Domain (FDTD) method may be used to electromagnetically model and simulate borehole radars and subterranean environments.

Opsomming

Suid-Afrika het 'n groot rykdom aan mineraalreserwes wat nog wag om ontgin te word. Dié reserwes kom egter voor in mineraalriwwe waar dit uiters gevaarlik en duur is om te myn. Hierdie is slegs twee van die redes hoekom boorgatradar in die afgelope tyd 'n baie belangrike navorsingsveld geword het vir die Suid-Afrikaanse mynindustrie. Hierdie radars moet werk in klip, wat 'n aantal elektromagnetiese karakteristieke bevat wat modellering en ontwerp van gepaste radars bemoeilik. Die doel van hierdie projek is om te demonstreer hoe die Eindige Verskil Tydgebied (EVTG) aangewend kan word om boorgatradars en ondergrondse omgewings elektromagneties te modelleer en te simuleer.

Acknowledgements

I would like to thank the following people for their support in my work on this thesis:

- Prof. David B. Davidson for his patience in guiding me in my work on my thesis.
- Dr. William F. Mitchell, of the Mathematical and Computational Sciences Division at the National Institute of Standards and Technology (NIST) of the US Department of Commerce, for his invaluable help with the Fortran OpenGL routines.
- My "Penthouse" colleagues for countless discussions on the basics of electromagnetics.
- Royce and Helen Caplinger for their keen interest in my work and their kind sponsorship towards the final publishing of this thesis.
- My parents, Dirk and Johanna Burger, for their personal and monetary support through six years of study.

Nomenclature

FDTD - Finite Difference Time Domain

ABC - Absorbing Boundary Condition

PML - Perfectly Matched Layer

3D - Three Dimentional

2D - Two Dimentional

1D - One Dimensional

DFT - Discrete Fourier Transform

FFT - Fast Fourier Transform

GUI - Graphical User Interface

GLUT - OpenGL Utility Toolkit

PEC - Perfect Electrical Conducter

FEM - Finite Element Method

MoM - Method of Moments

Contents

List of Figures

Chapter 1

Introduction

1.1 An Overview of the *Deepmine* Project

South Africa is undoubtedly one of the richest geologic regions in the world, and mining practices have grown and modernized in South Africa for a number of years. The mining industry is currently mining down to depths of approximately 3.5km, but the reserves that are still available to be mined, at these depths, will be exhausted in the not too distant future. According to Trickett *et al.*, [1], it is therefore imperative that technology enabling mining at ultra-deep levels have to be pursued. This statement is based in part on the fact that gold reserves, that are estimated to be deposited between depths of 3km to 5km, could well be equal to the amount of gold that was mined from the reefs of the Witwatersrand basin during the past hundred years, [2]. This manner of thinking led to the creation of the *Deepmine* project in March of 1998.

The goals of the *Deepmine* project are to investigate all aspects of ultra-deep mining. A few of the more prominent fields of interest are:

- Occupational health and safety.
- Delineation and characterization of geological structures.
- Mine layout and mining methods, shaft sinking and design.
- Transport systems, energy systems and process control.

Knowledge of geological features and structures would enable the careful planning of a mine's layout, which would in turn minimize the probability of seismic events such as rockbursts, Methane gas explosions or the breach of high pressure water fissures, all of which could cause very serious accidents or possibly loss of life.

It was, therefore, decided that a borehole radar should be developed for the estimation of physical subterranean features in ore-bearing rock bodies. Such a radar would be an invaluable tool in mine planning, and should be able to show a number of features in an ore-bearing reef, such as:

- Faults,
- Water fissures and
- "Rolls" [1]

1.2 Thesis Goals

At this stage of development of the *Deepmine* borehole radars, research has focused on the design of the radar and associated electronics for the system, and the importance of finding or developing a numerical tool for the simulation of such an environment has come to the fore lately.

The goal of this thesis is to show how the FDTD method, coded in Fortran, may be used to create a simulation tool, which could model the following characteristics of a borehole radar system and environment:

- Resistively loaded dipole antennas.
- Dispersive media.
- Stratified media.

A simplified depiction of a borehole radar environment is presented in figure 1.1.

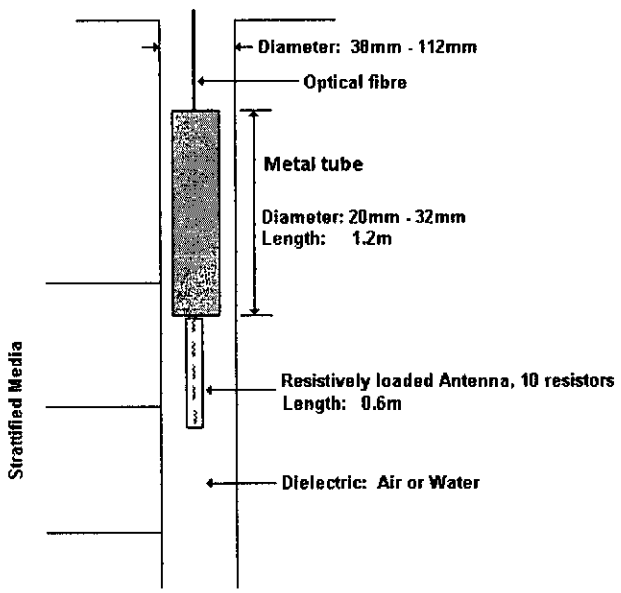


Figure 1.1: Graphical description of the *Deepmine* radar environment

The FDTD modelling technique has been in use for a number of years now and offers a number of features and possibilities for modelling and simulation, that other numerical modelling tools do not. The formulations that are needed to model the physical

[1] "Roll" is the industry slang for an undulation in the gold-bearing Ventersdorp Contact Reef.

phenomenon of interest to this thesis, all exist at least in basic form in the available literature, which was carefully surveyed and documented by Shlager in [3]. The main contribution of this thesis will be through the implementation and evaluation of these numerical models for this specific problem. The resulting code could assist in evaluating and improving the current system in the following ways:

- The pulse behaviour of current and possible future antennas could be investigated.
- The pulse behaviour of the system in a dispersive medium, such as water, could be investigated.
- The directionality of the current antenna could be characterized in a stratified media, by looking at the propagation of pulses into a stratified medium.
- The radiation impedance of an environment with various dielectrics in irregular formations around the antenna could be investigated, with the information being used towards designing an antenna better matched to such environments.

A secondary objective is to illustrate the usefulness of OpenGL as a visualization tool for data generated in Fortran. It is, however, a secondary objective and the work will thus be presented as an Appendix.

Chapter 2

Introduction to the Finite Difference Time Domain

2.1 The Yee Algorithm

2.1.1 Basic Ideas

To model a system where electromagnetic propagation takes place it is imperative that a technique be found that solves Maxwell's equations, which are given in the Appendix section A.1. Kane Yee subsequently introduced the FDTD for the first time in 1966, [4]. Yee's algorithms were based on Maxwell's curl equations and are based in the time domain. The Yee algorithms provided a basic and very robust modelling method which is still used because of the following attractive features, [5]:

- The Yee algorithm solves for both the electric and magnetic fields in time and space, using the coupled Maxwell's curl equations rather than solving for the electric field alone with a wave equation. This solution provides modelling capabilities for a much wider range of physical problems, which contain specific features in both the electric and magnetic fields.
- The Yee cell, shown in figure 2.1, centers $\overline{E}$ and $\overline{H}$ components in 3D space so that every $\overline{E}$ component is surrounded by four circulating $\overline{H}$ components, and each $\overline{H}$ component is surrounded by four circulating $\overline{E}$ components. This formulation presents a simple structure representing the laws of Ampére and Faraday, which is interlinked in time and space.
- The resulting finite-difference expressions are central in nature and are second-order accurate.
- Continuity across an interface of dissimilar materials provides an easy method for simulation of inhomogeneous media.
- The location of the $\overline{E}$ and $\overline{H}$ field components in the grid implicitly enforces the two laws of Gauss, thus rendering the Yee cell divergence free in a source-free space.

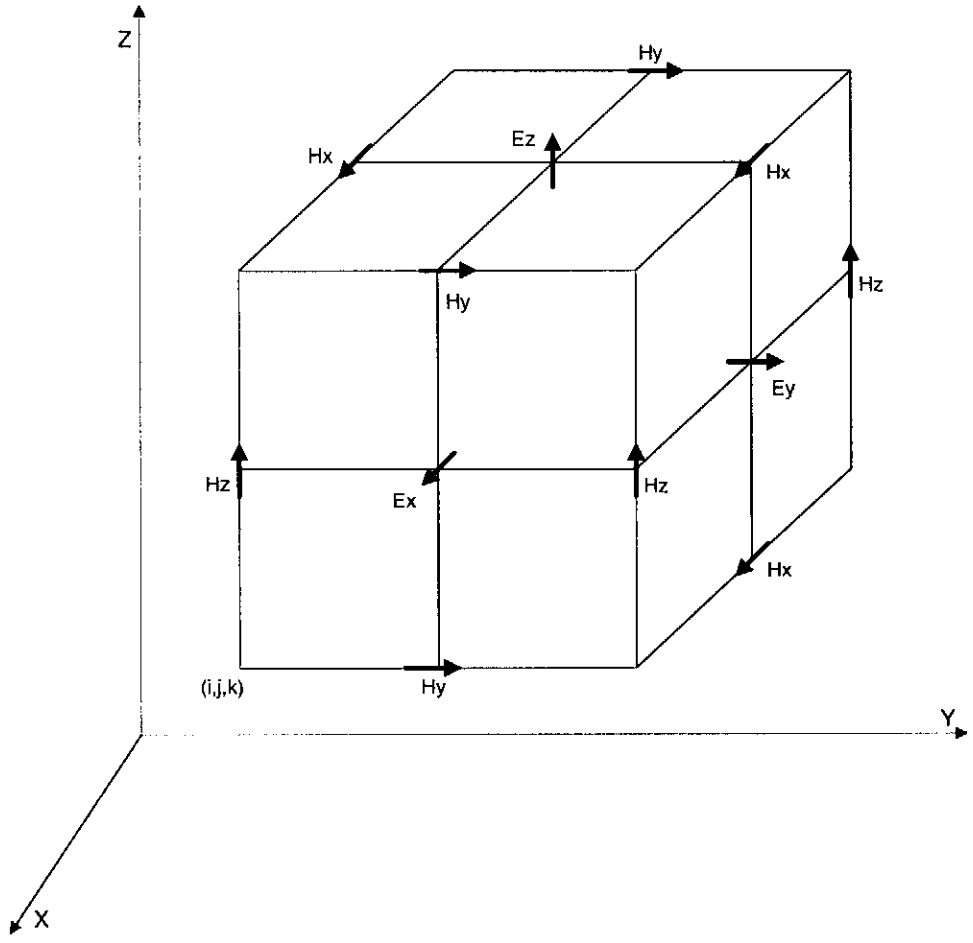


Figure 2.1: Graphical description of the Yee cell

2.1.2 Finite Differences and Notation

The notation that is used in this thesis was also introduced by Yee and is presented in Taflove [5], but is repeated here for convenience.

Any specific point in space is represented as:

$$(i, j, k) = (i\triangle x, j\triangle y, k\triangle z) \tag{2.1}$$

where $\triangle x$, $\triangle y$ and $\triangle z$ represent the lattice spatial dimensions in the x, y and z directions respectively. The integer values i, j and k then allow us to denote a function of space and time as:

$$u(i\triangle x, j\triangle y, k\triangle z, n\triangle t) = u^n_{i,j,k} \tag{2.2}$$

where $\triangle t$ is the time increment and n is again an integer.

Yee used the theory of central-differencing to create expressions for the time and spatial derivatives that are required in Maxwell's curl equations. The differences were approximated with the point of interest, either time or spatial, being approximated by the central difference of the two points one halfspace on either side of it. This approach results in

spatial and time difference equations of the following format:

$$\frac{du}{dx}(i\triangle x, j\triangle y, k\triangle z, n\triangle t) = \frac{u^n_{i+\frac{1}{2},j,k} - u^n_{i-\frac{1}{2},j,k}}{\triangle x} \quad (2.3)$$

$$\frac{du}{dt}(i\triangle x, j\triangle y, k\triangle z, n\triangle t) = \frac{u^{n+\frac{1}{2}}_{i,j,k} - u^{n-\frac{1}{2}}_{i,j,k}}{\triangle t} \quad (2.4)$$

The halfspace and halftime points, that are required in these equations, conveniently fit in with Yee's idea of a *leapfrog* algorithm. The leapfrog idea implies that all the electric field values and all the magnetic field values may be updated in turn, which in effect means that the corresponding electric and magnetic field points are removed from each other by one spatial halfspace and by one half timestep. The update equations for the E_x and H_x field components in freespace are presented in equations 2.5 and 2.6 as an example of the resulting field equations.

$$E_x|^{n+1}_{i,j,k} = E_x|^n_{i,j,k} + \frac{\triangle t}{\epsilon_0}\left(\frac{H_z|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k} - H_z|^{n+\frac{1}{2}}_{i,j-\frac{1}{2},k}}{\triangle y} - \frac{H_y|^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} - H_y|^{n+\frac{1}{2}}_{i,j,k-\frac{1}{2}}}{\triangle x}\right) \quad (2.5)$$

$$H_x|^{n+\frac{1}{2}}_{i,j,k} = H_x|^{n-\frac{1}{2}}_{i,j,k} + \frac{\triangle t}{\mu_0}\left(\frac{E_y|^n_{i,j,k+\frac{1}{2}} - E_y|^n_{i,j,k-\frac{1}{2}}}{\triangle z} - \frac{E_z|^n_{i,j+\frac{1}{2},k} - E_z|^n_{i,j-\frac{1}{2},k}}{\triangle y}\right) \quad (2.6)$$

These equations may be extended to include all the other equations required for 3D propagation by simple perturbation of the subscripts.

2.2 Numerical Stability

The timestepping equations of the FDTD require a specific relation between the size of the spatial cells and the length of the time increment. This relation would help avoid the undesirable situation where the computed results spuriously increase without limit as the algorithm steps through time. The resulting criteria, known as the Courant limit, that is necessary but not sufficient for numerical stability may be expressed as follows:

$$\triangle t \leq \frac{1}{c\sqrt{\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2} + \frac{1}{\triangle^2}}} \quad (2.7)$$

where c denotes the speed of light in freespace.

The Courant condition essentially defines the timestep of the simulation to be equal to or smaller than the maximum length dimension (L) of the spatial cell divided by the speed of wave propagation in the medium. The mathematical description given above is for a 3D cell. When a 2D or 1D problem is solved the appropriate dimensions may simply be omitted from the equation. A graphical description of the distances involved for the 1D, 2D and 3D Courant condition is given in figure 2.2.

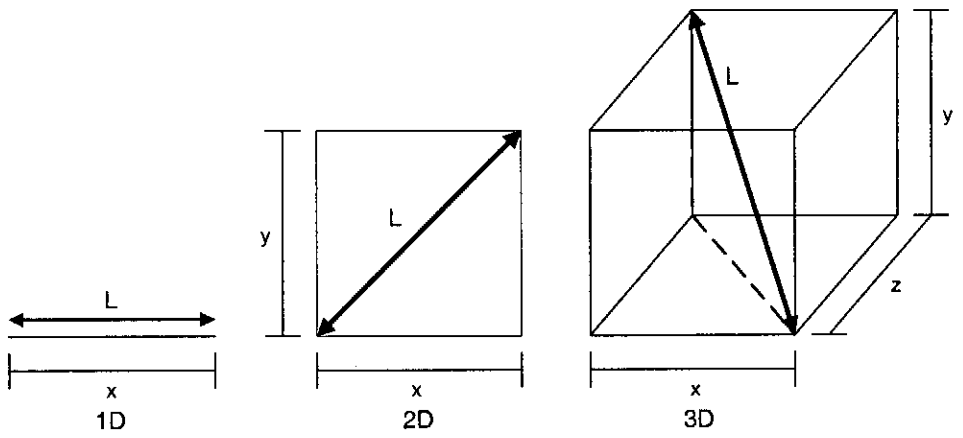


Figure 2.2: Graphical description of the Courant condition

It is important to note the distinction that the equality of equation 2.7 relies on the speed of light *in the medium.* In a dielectric it is therefore important to find the limit for $\triangle t$ with c replaced by the speed of light in the medium v, which is defined as:

$$v = \frac{c}{\sqrt{\epsilon_r \mu_r}} \tag{2.8}$$

where ϵ_r and μ_r is the relative permittivity and relative permeability of the dielectric.

2.3 Finite Difference Time Domain Field Formulations

On formulating a FDTD problem, one is immediately confronted with the choice of using a *total field* or a *scattered field* formulation.

In the *scattered field* formulation only the scattered field propagates through the medium, as the name and the formulation of equations 2.9 and 2.10 suggests.

$$E = E^{incident} + E^{scattered} \tag{2.9}$$
$$H = H^{incident} + H^{scattered} \tag{2.10}$$

As Maxwell's equations are linear in nature, the basic curl equations may be written as sum and product terms of the separate fields:

$$\nabla \times \left(E^{inc} + E^{scat}\right) = -\mu \frac{d\left(H^{inc} + H^{scat}\right)}{dt} \tag{2.11}$$
$$\nabla \times \left(H^{inc} + H^{scat}\right) = \epsilon \frac{d\left(E^{inc} + E^{scat}\right)}{dt} + \sigma \left(E^{inc} + E^{scat}\right) \tag{2.12}$$

The incident waves or source pulses are specified analytically for the entire region of interest, and are then used to calculate the scattered fields in the simulation region. The main advantages of using the scattered field formulation are (ref. [6]):

- The incident wave or pulse is not prone to numerical dispersion, whilst propagating in the problem space, because it is specified analytically.
- Only the scattered fields have to be absorbed by the problem space boundary. The scattered fields are normally of much lower amplitude than the incident wave, which helps to absorb them much more readily with a boundary condition.

The *total field* formulation, on the other hand, possesses some attractive characteristics for use in this project. The main advantages to using the total field formulation are:

- The total field propates through the medium, and provides better insight into the propagation aspects of the medium and interaction with structures of interest in the problem space.
- Discrete sources, like antennas, with specific radiation characteristics are much simpler to implement and visualize than with the scattered field formulation.

As both formulations are completely valid in formulating specific phenomenon of interest to this project, it does not make a difference from a mathematical point of view which formulation is chosen to achieve the goals of this project. What does, however, make a difference is the ease of implementation and insight that could be gathered from viewing the fields propagating in one or the other implementation. Although the scattered field formulation would be ideal for computing the reflections from different layers in a stratified media environment, such as layers of rock, the emphasis in this thesis falls on the study of the behaviour of an antenna placed in a borehole. It was subsequently decided that the total field formulation provides more attractive features and advantages applicable to the project and was chosen as the formulation to use.

2.3.1 Scattered Field Formulation Conversion to Total Field Formulation

The development of a total field formulation FDTD code, from scratch, is a timeconsuming process and did not fall within the scope of this thesis. Alternatively, a scattered field formulation code (originally published in [6] and modified by D.B. Davidson) was converted into a total field formulation code. The original code contained functions for basic freespace propagation, with the area of simulation being bounded by a second order Mur boundary condition.

According to [6, p. 13] a scattered field formulation FDTD code can easily be converted to a total field formulation code by simply setting the incident field to zero and adding the incident field and initial conditions to the scattered field components. This statement was confirmed by derivation, and can also be confirmed by inspection of equations 2.11 and 2.12. If the incident field components in each of these equations are set to zero and the incident fields are added to the scattered field components, the formulation defaults to the total field formulation of Maxwell's equations.

2.4 Absorbing Boundary Conditions

One of the basic problems that is encountered in the implementation of a FDTD code is that freespace radiation problems need to be analyzed, while computers never satisfy the computational requirements for an infinitely large computational space. The solution to this problem is to implement a boundary condition that only allows propagation in one direction, i.e. outwards. Such boundary conditions enable the simulation of freespace problems if they absorb enough of the energy that is radiated outward. The two prominent ABC's that are used in FDTD codes are the Mur ABC of [7] and the Berenger PML of [8].

2.4.1 The Mur Absorbing Boundary Condition

The Mur boundary condition is an implementation of the one-way wave equations that were first developed by Engquist and Majda, [9], for Cartesian coordinate systems. Their theory rests on the factoring of the partial derivative operators of the 3D wave equation:

$$\frac{d^2U}{dx^2} + \frac{d^2U}{dy^2} + \frac{d^2U}{dz^2} - \frac{1}{c^2}\frac{d^2U}{dt^2} = 0 \tag{2.13}$$

The Mur boundary condition was first published in 1981, but is still used today. The main features of the Mur ABC are that it is relatively easy to implement and has second order accuracy.

2.4.2 The Berenger Perfectly Matched Layer

With the advent of anechoic chambers with dynamic ranges of the order of 70dB it has become important to use a boundary condition with similar reflection features. According to Taflove, [5, p. 181], the existing ABC's, like the Bayliss-Turkel, Engquist-Majda, Higdon and Liao, only provide effective reflection coefficients in the range of 0.5% to 5%, which is clearly not sufficient.

Berenger recently (1994) proposed a new method of absorbing radiation at a boundary, [8]. His method suggests the splitting of electric and magnetic field components into separate fields in the region of the absorbing boundary, and then assigning losses to each individual field component. Initially the paper only implemented the perfectly matched layer for a 2D environment and reported "orders of magnitude" improvement over the then available ABC's. Berenger's work was subsequently validated and extended to three dimensions by Katz *et al.* in [10], and independently by Berenger himself in [11]. The 3D implementation of the Berenger PML is reported to increase the maximum dynamic range of FDTD problem space to greater than 100dB.

Other improvements on the original Berenger PML have been made by Akleman and Sevgi, [12], who introduced a technique which doubles the effective cell size in the absorption region and in doing so achieves significant improvements in the simulation of

3D structures that exhibit complex wave phenomenon, and Uno *et al.*, [13], who applied Berenger's PML to the termination of a dispersive medium.

Unfortunately the Berenger PML has some drawbacks. As the splitting of field components by the Berenger PML is essentially a non-Maxwellian mathematical model, it leaves questions about whether such materials are physically realizable and the FEM applicability of such absorbing materials. The Berenger PML subsequently served as the basis for two other mathematical techniques that attempt to address these questions, namely the uniaxial anisotropic absorber and the stretched coordinate formulation, which are discussed in [14, paragraphs 5.3, 5.4].

The stretched coordinate system was originally introduced by Chew and Weedon in [15] and by Rappaport in [16], and although the stretched coordinate system is easier to manipulate and understand than the original split field formulation, it must be said that it is still not a physically realizable medium. The anisotropic PML was subsequently introduced by Sacks *et al.* in [17]. It is uniaxial and is composed of both electric and magnetic permittivity tensors, and performs as well as Berenger's PML while avoiding the splitting of fields.

Chapter 3

FDTD Modelling of Impedance Loaded Dipoles

3.1 Introduction

In the mining environments, where the *Deepmine* radars have to operate, the physical dimensions of antennas are limited by the geometry of the holes that are used to deploy the radar. The various physical constraints were met by the implementation of an impedance loaded dipole, which was implemented using chip resistors and metallic strips printed on a dielectric substrate.

Studies that were done to improve the radiation characteristics of this antenna[1] focused on the antenna that is depicted in figure 3.1. This antenna consists of two identical arms, each consisting of ten metallic strips, with nine equally spaced gaps for impedance elements. The feed and end elements are half the length of the other elements.

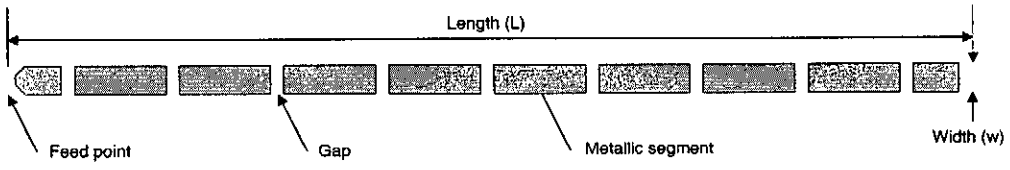


Figure 3.1: One arm of an impedance loaded dipole

This chapter investigates the FDTD modelling of such antennas.

3.2 FDTD Modelling of Irregular Structures using Fine Geometric Models

One of the questions that often arise in any grid-based modelling technique is what to do when the structures of interest do not conform to the grid. Physical structures do not

[1]Master's thesis project of Sebastian Keller, [18]

conform to a regular grid for a number of reasons, e.g. scale differences or curved surfaces in a regular mesh. Two possible solutions exist for this problem:

- Refine the modelling technique and grid to be able to handle the irregular structures.
- Maintain a regular grid, while modelling irregular features of a problem on a sub-cellular scale, known as a fine geometric model.

Both options have some attractive features. Option 1 would maintain geometrical fidelity while providing second order accuracy everywhere in the space grid. The problems with this approach are unfortunately prohibitive. Computer resource requirements tend to become very large and as a result simulation times also become very lengthy, and the process of mesh generation becomes a highly specialized and timeconsuming task itself.

While option 1 provides more accurate results than option 2, according to [5, p. 282], option 2 relaxes the computer resource and time requirements considerably, while maintaining a high degree of accuracy. It relies on a regular grid, which makes it easy to implement from a grid generation perspective, and requires little extra mathematical formulation to implement several types of irregularly structured objects.

The basic principle behind the fine geometric model is the principle of contour-path modelling. The new update equations for the fields around an irregular structure are derived from the integral expressions for the laws of Ampére and Faraday, presented in the Appendix equations A.2 and A.4. These fundamental field equations are implemented on an array of electrically small and spatially orthogonal contours, which link together in the manner of links in a chain and provides a geometrically satisfying description of the interaction between the two laws. [2]

3.3 Modelling Strip Antennas

Although it is possible to implement thin metal strips in an FDTD code, it is not necessary, as wire elements with the same radiation characteristics as that of the strips may be used for simulation purposes. The equivalent radius principle involved in transforming a strip to an equivalent wire may be found in [19], and is stated quite simply as:

$$a_{eq} = \frac{w}{4} \tag{3.1}$$

with a_{eq} the equivalent radius of the wire antenna and w the width of the metallic strip.

[2] Refer to [5, p. 73, figure 3.3]

3.4 Modelling a Thin Wire Resistively Loaded Dipole Antenna

3.4.1 The Thin Wire Model

The theory of fine geometric models is now applied to a thin PEC wire. The following assumptions are made in the process of deriving update equations for the fields around the wire:

- The wire is z-directed.
- The wire has radius less than half of any FDTD spatial cell dimension.
- The scattered looping field components H_y and H_x and the radial electric field component E_x vary as $\frac{1}{r}$ near the wire, where r is the radius of the wire.
- The $\frac{1}{r}$ scattered field singularity behaviour dominates the respective fields.
- The field components in the center of the contours and areas of integration represent the average value of the field values over the entire interval of integration.

Using these assumptions and the laws of Ampére and Faraday the thin wire update equations are easily derived. The derivations for the H_y field component update equations are given in the Appendix, A.3, and the update equations themselves in A.3.1.

3.4.2 Lumped Element Models

Deltagap Voltage Source

The two most prominent methods for feeding a wire antenna in a FDTD application is the "deltagap" model, as described by Watanabe and Taki in [20], or the "one-cell gap" model, as described by Kunz and Luebbers in [6]. Watanabe and Taki analyze both techniques in their paper and come to the conclusion that the deltagap model gives more accurate results than the one-cell gap model. The feeding structure for the antennas in this project was therefore implemented as a deltagap model.

The basic idea behind the deltagap model is to expand the update equations for the thin wire antenna, as given in section A.3.1, to include a source term. This means that the thin wire approximation in the FDTD grid is still used but that an infinitesimal gap is included in the wire. This gap is then fed with a resistive voltage source or a pure field source.

The update equations for a deltagap voltage source are presented in the Appendix, section A.3.2

Deltagap Resistor

The deltagap feed model is also the basis for the implementation of the lumped element resistor in the resistively loaded antenna. A concise explanation of the implementation of a *numerical* resistor in contrast to a *physical* resistor is given by Piket-May in [21].

The implementation of a physical resistor dictates that an entire cell be specified as a resistor with the following relation in mind:

$$R = \frac{\rho L}{A} \tag{3.2}$$

This equation realizes the resistor by specifying a resistivity, ρ, for the entire cell of length L and cross-section A.

The method that Piket-May advocates is to insert a numerical lumped element in an extension to the FDTD. The basic concept is that Maxwell's equation for Ampére's law is augmented by a term related to the current in the lumped element, and may be rewritten as:

$$\nabla \times \overline{H} = \frac{d\overline{D}}{dt} + \overline{J}_L \tag{3.3}$$

In applying the new formulation for Ampére's law it is assumed that:

- The lumped element is located in freespace.
- The lumped element is z-orientated in the grid.
- The local current density in the lumped element is related to the total element current, I_L, as:

$$J_L = \frac{I_L}{\triangle x \triangle y} \tag{3.4}$$

  where I_L describes the conduction current in the element.

- I_L is assumed to be positive in $+z$ direction.

In the case of the resistor I_z and J_L are assumed to be of the form:

$$I_z = \frac{\triangle z}{2R}\left(E_z|_{i,j,k}^{n+1} + E_z|_{i,j,k}^{n}\right) \tag{3.5}$$

$$J_L = \frac{I_z|_{i,j,k}^{n+\frac{1}{2}}}{\triangle x \triangle y} \tag{3.6}$$

where R is the resistance of the resistor to implement, given in Ohm.

This formulation was used to derive the new update equation for the z-directed electric field component in the center of the resistor. The updates for the magnetic field components in the cell, where the resistor is placed as part of the wire, stay the same as for the deltagap source. The source term is, however, replaced with the electric field term in the center of the resistor. This, in effect, means that an infinitely small resistor has been placed between two sections of the wire.

The update equations for the deltagap lumped element resistor are presented in the Appendix, section A.3.3.

Deltagap Capacitor

The deltagap capacitor formulation is similar to the formulation for the deltagap resistor. The difference in the derivation lies in the specification of the current in the lumped element.

The assumptions for the current direction and density are held as they were for the resistor with the description of the current through the capacitor being changed to:

$$I_z = \frac{C\triangle z}{\triangle t}\left(E_z|_{i,j,k}^{n+1} - E_z|_{i,j,k}^{n}\right) \tag{3.7}$$

$$J_L = \frac{I_z|_{i,j,k}^{n+\frac{1}{2}}}{\triangle x \triangle y} \tag{3.8}$$

where C is the capacitance of the capacitor to implement in Farad.

The update equations for the deltagap lumped element capacitor were derived and are presented in the Appendix, section A.3.4.

3.5 Impedance Loaded Antenna Code Verification

The code implementing the theory and update equations involved in modelling impedance loaded flat dipoles were tested by comparing the input impedance of the antenna, calculated with the FDTD code and MATLAB, to the input impedance computed by FEKO[3].

3.5.1 Obtaining Antenna Input Impedances from the FDTD

The formulation used in obtaining the input impedance of an antenna from the FDTD was taken from [5, p. 433].

Taflove describes how the input impedance of an antenna may be found from Fourier transforms of the voltages and currents through the feedpoint of the structure of interest. The derivation requires a couple of mathematical interpolations:

- The feedpoint is assumed to fall on an E-field point, or in impedance terms, on a voltage point. It is therefore necessary to take the geometric average of the currents directly adjacent to this point.
- The E-fields and H-fields, or voltages and currents in terms relating to impedances, are not sampled at the same time and the time ambiguity must be removed by multiplying the expression for voltage by the factor $e^{-j\omega\frac{\triangle t}{2}}$.

The resulting equation, and its derivation, which is used to calculate the input impedance of a dipole is presented in the Appendix, section A.4.

[3]FEKO is a hybrid Method of Moments (MoM) code, supported by EM Software and Systems, Technopark, Stellenbosch

As the input impedance of an antenna is plotted against frequency, a Fourier transform is needed in the process. Initially the FDTD was only used to sample the time-domain waveforms with a MATLAB FFT being used to calculate the input impedance. One of the problems with this approach was that is was hard to guarantee that certain frequency components would be available in the output. A DFT was therefore implemented in the FDTD code to solve this problem. Refer to Appendix chapter B for an explanation of the theory behind the implementation of a DFT and the merits of a DFT vs. FFT implementation.

3.5.2 Wu-King Resistively Loaded Dipole Simulation Results

Figures 3.2 and 3.3 show comparative simulation results for two Wu-King profile resistively loaded dipoles, originally designed and simulated by Keller, [22]. Each arm of the dipole has the following specifications:

- Length 150mm.
- Width 6mm (equivalent radius 1.5mm).
- 9 gaps for equally space resistors (The first and last segments are half the length of the other segments).

The two load profiles simulated here are for resistor values (feed to end):

- 15 - 20 - 27 - 36 - 56 - 100 - 200 - 680 - 11k ohm
- 46 - 52 - 61 - 72 - 88 - 113 - 158 - 265 - 795 ohm

Each figure shows a graph comparing results from FEKO, generated by [22], and the FDTD code of this project.

3.5.3 Wu-King Capacitively Loaded Dipole Simulation Results

Figures 3.4, 3.5, 3.6 and 3.7 show comparative simulation results for two exponential profile capacitively loaded dipoles, originally designed and simulated by Keller, [22].

The geometric dimensions of the dipoles are the same as for the resistively loaded dipoles that were simulated in the previous section.

The two load profiles simulated here are for capacitor values (feed to end):

- 10.0p - 4.7p - 2.7p - 2.2p - 1.5p - 1.5p - 1.2p - 1.0p - 1.0p Farad
- 4.7p - 2.2p - 1.8p - 1.5p - 1.2p - 1.2p - 1.0p - 1.0p - 1.0p Farad

As with the resistively loaded dipoles, each figure shows a graph comparing results from FEKO, generated by [22], and the FDTD code of this project.

3.5.4 Comments on Simulation Results

Figures 3.2 and 3.3, that compare the resistively loaded antenna input impedances, compare extremely well, which validates the FDTD formulation for resistors. Resistors are electrically simple elements to model, because of the fact that they are linear devices and do not store energy in any form.

The results for the capacitively loaded antennas, figures 3.4 through 3.7, do not agree as well as for the resistively loaded case, and could be explained by the difference of opinion on how the capacitor should be modelled in the FDTD. Piket-May *et al.*, [21], models the capacitor with a relation that spans one timestep, while Sui *et al.*, [23], describes the capacitor with a model spanning two timesteps. This thesis implemented the approach of Piket-May *et al.* and did not further investigate the effect of adding another timestep to the capacitor model.

The FDTD does, however, model the capacitor sufficiently well to predict the trends and magnitudes of the input impedances for the two antennas to an acceptable degree of accuracy.

It must also be noted that although MoM solutions to these simulations are available and work well, the goal of this thesis was to create and validate a code that would be able to simulate the antenna of interest in a non-homogeneous environment, and this is something that the FDTD handles much more readily than the MoM.

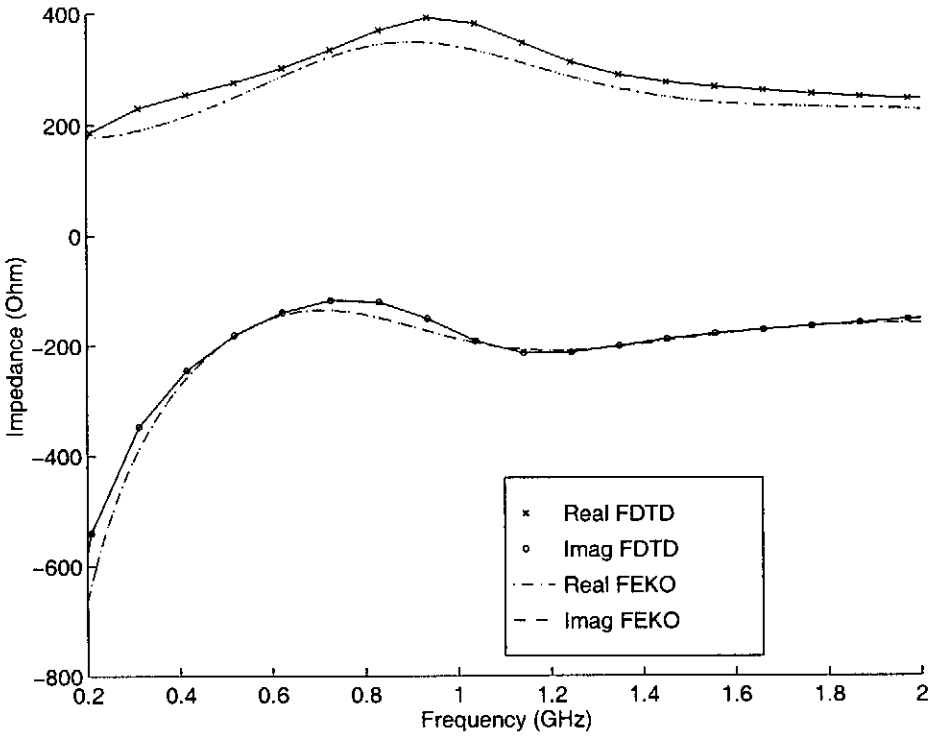


Figure 3.2: Resistively loaded input impedance comparison - Profile 1

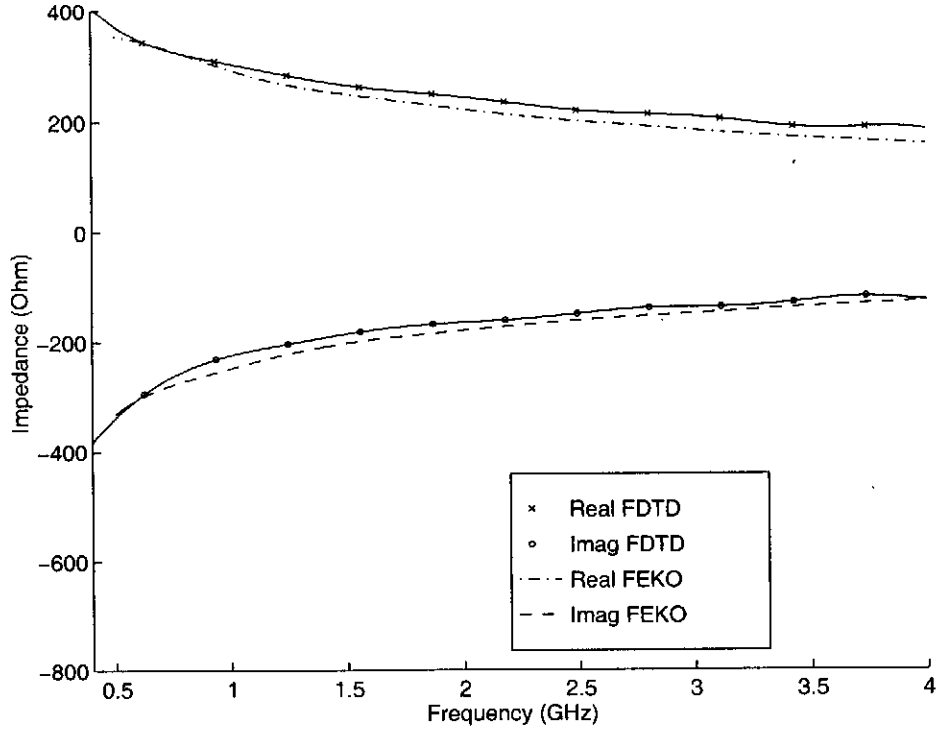


Figure 3.3: Resistively loaded input impedance comparison - Profile 2

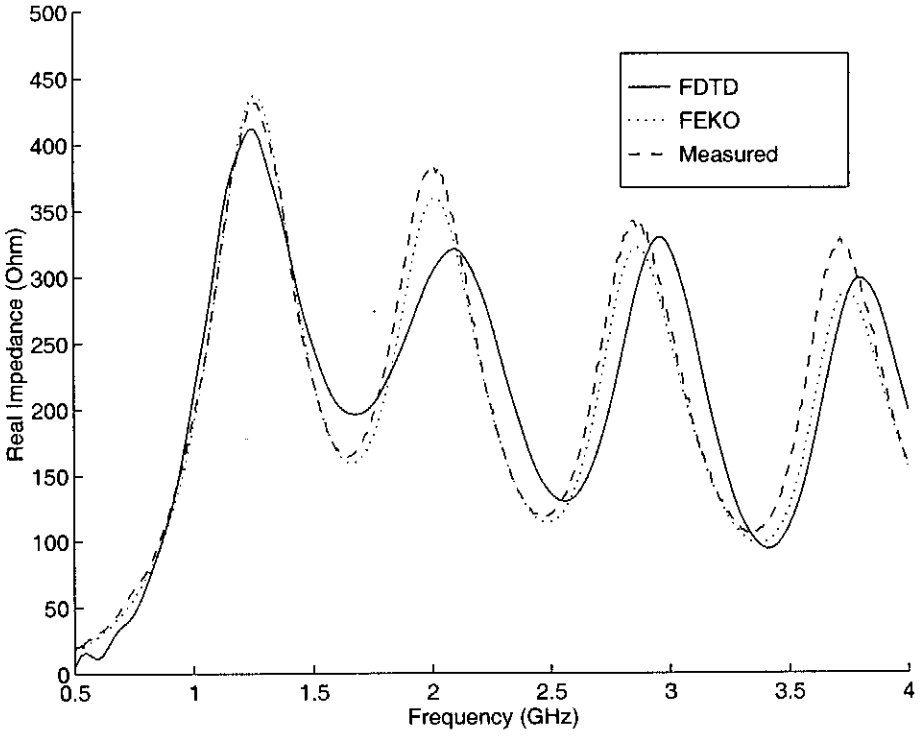


Figure 3.4: Capacitively loaded input impedance comparison (real component) - Profile 1

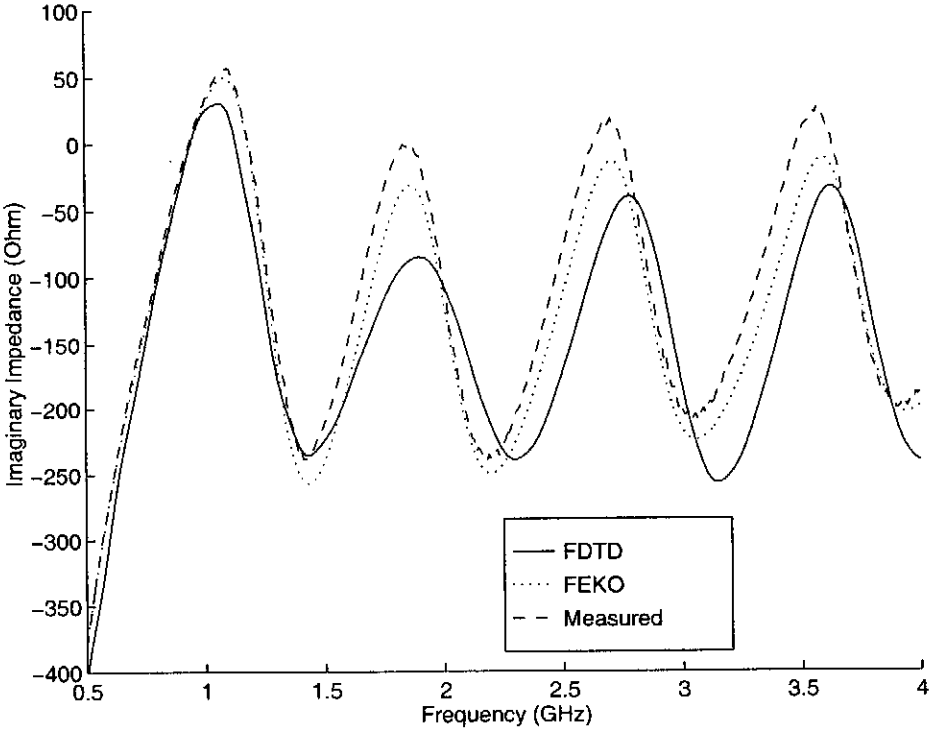


Figure 3.5: Capacitively loaded input impedance comparison (imaginary component) - Profile 1

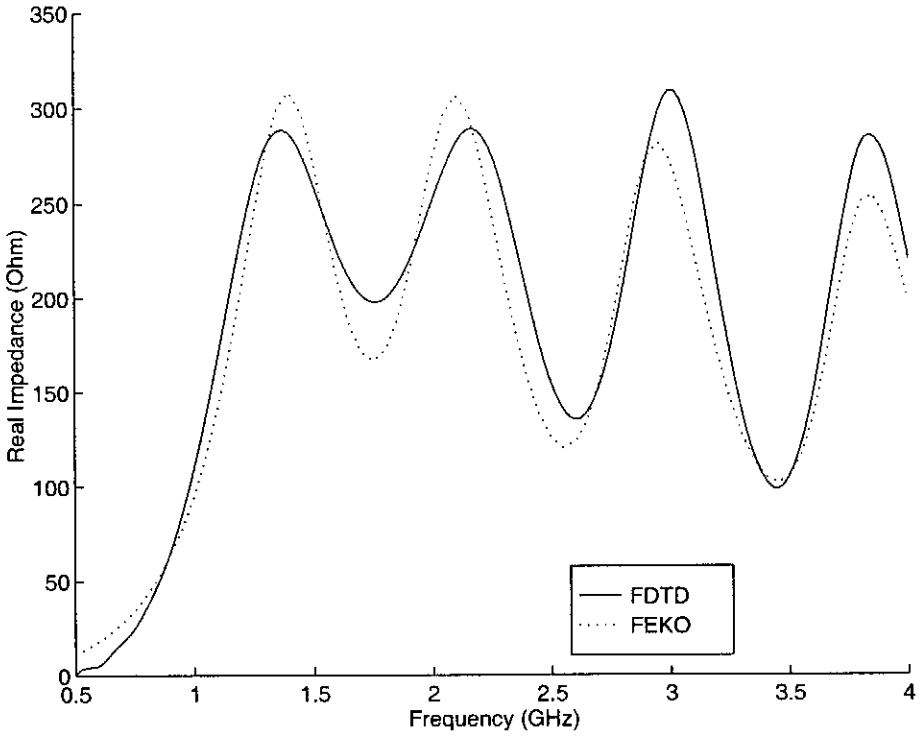


Figure 3.6: Capacitively loaded input impedance comparison (real component) - Profile 2

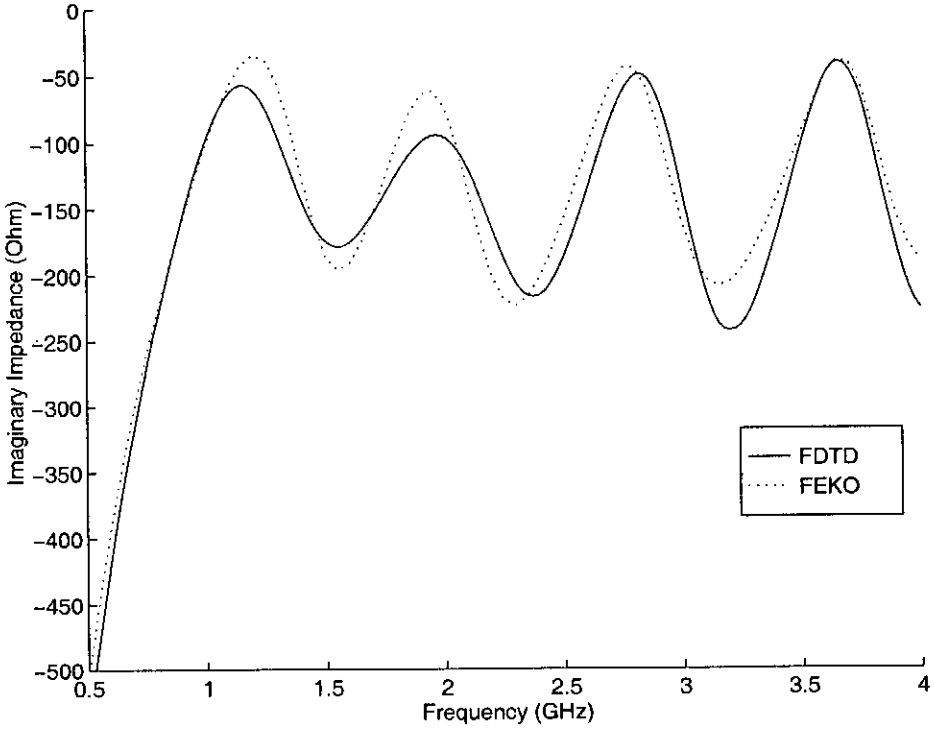


Figure 3.7: Capacitively loaded input impedance comparison (imaginary component) - Profile 2

Chapter 4

FDTD Radiation Patterns

4.1 Introduction to FDTD Near-Field Measurements

Part of the *Deepmine* project focuses on the improvement of the radiation patterns of the antennas that are used in the radar, and it is therefore imperative that a numerical tool for these studies have to be able to produce far-field patterns for the antennas that are being simulated.

Conceptually it is possible to use a sampled near-field, with the samples in phasor format, as a source term to calculate far-field patterns via radiation integral theory, as presented in [24, paragraph 12.12]. A number of questions arise when this process has to be implemented using the FDTD method, with the fields evolving as explicit functions of time, such as:

- How and where is the near-field defined?
- How may the near-field be sampled in phasor format?
- How may the sampled near-field be transformed into a far-field pattern?

This chapter will address these issues and some of the implementation issues that arise with them.

The answer to the first question is defined in a number of texts, and a few of them define the limits of the near-field by the same reasoning. One of the more popular definitions state that the near-field region is defined as extending from start of the evanescent region to an arbitrary distance from the antenna, normally $\frac{2D^2}{\lambda}$, where D is the largest dimension of the structure of interest. The boundaries between the different electromagnetic field regions are shown in figure 4.1.

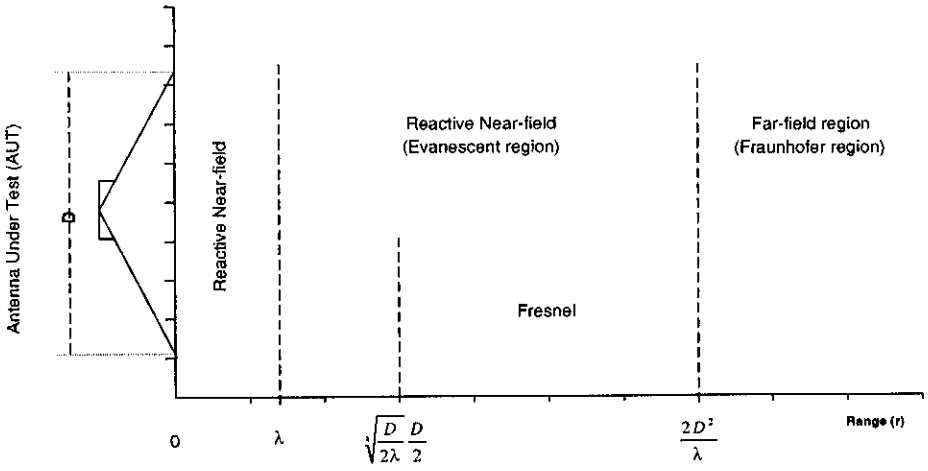


Figure 4.1: Exterior field boundaries of a radiating antenna

4.2 Transformation Methods for Near-field to Far-field Data

4.2.1 Two Dimensional Fourier Transform into an Angular Spectrum.

The transformation of a phase front into an angular spectrum, with a 2D Fourier transform, is often called a *near- to far-field transformation.* This is actually incorrect, according to [25, p. 44], as the transform input and output are at the same location, where the measurements were made. The transform converts only between a phase front and an angular spectrum (i.e., the spectrum of directions in which the wave is traveling) at the same position in space. The transformation thus results in the equivalent of a far-field pattern because the radiated near-field energy components are always traveling in a straight line at any distance from the antenna under test.

4.2.2 Equivalent Dipoles Method

Taflove explains the necessary theory to implement Green's functions and sampled phasors in a FDTD code for the calculation of far-field patterns in [5, chapter 8], but this is specialized work that is not central to the FDTD method, and FEKO may be used to generate far-field patterns instead. The FEKO user manual, [26], explains how each sampled near-field point, electrical or magnetic, is replaced by an equivalent dipole. These dipoles could then be used as legitimate sources of electromagnetic radiation that interact to result in a far-field pattern, which approximates the true far-field pattern to a high degree of accuracy. If both a magnetic and electric near-field is specified, at the same point in space, a direction of propagation is in effect also specified with a resulting Poynting vector.

4.3 Empirical Experimentation

FEKO was used to generate near-field data for a resistively loaded dipole antenna[1] which was then used as a source to calculate far-field data. Investigations into the accuracy of a far-field pattern, calculated with a near-field scan as source, were done by [27] and empirically investigated in this project. The investigations, applicable to both transformation techniques, indicate that if only one sampled plane is used as the source the calculated far-field pattern would be accurate up to an angle of θ, described by figure 4.2.

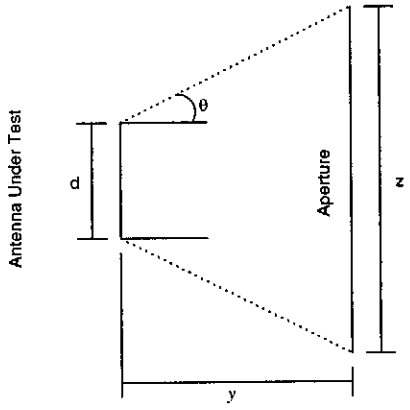


Figure 4.2: Accuracy estimation for near-field to far-field transformations.

θ may be expressed as follows:

$$\theta = \arctan(\frac{z-d}{2y}) \tag{4.1}$$

This restricts the applicability of just one plane of near-field data when an accurate description of the far-field behaviour of a structure is needed in all directions. Near-field planes should therefore be sampled in a box-like fashion around the scatterer or radiator of interest. Radiation would be sampled in all directions in the near-field and an accurate transformation of near-field to far-field data would be possible.

An intuitive analysis of this concept would suggest that the discretization of such sampled grids would also play an important part in the transformation process. Empirical studies were done to evaluate different sampling and transformation criteria. The evidence gathered from these investigations suggest the following:

- A cube should be sampled around the structure of interest if all directions of the far-field pattern are to be approximated accurately.
- Finer discretization of sampled grids improves the far-field data.
- Sampling very close to the structure of interest, as opposed to sampling further off, also improves the far-field radiation pattern

The evidence that supports these statements is presented in the Appendix, chapter C.

[1] Model obtained from [22]

4.4 Using FEKO and the FDTD Method to Calculate Far-fields

The way to use FEKO in this regard is to specify source planes consisting of electric and magnetic field phasors. These phasors are then converted to the equivalent electric and magnetic dipoles of section 4.2.2 by the FEKO AP-card, [26, AP-card], which are then used to calculate the required far-field patterns in conjunction with the FEKO Green's functions for free-space.

The implementation of phasor sampling in the near-field requires a Fourier transform, which is explained in the Appendix, chapter B. The resulting DFT formulation and code fragments from Taflove, [5, paragraph 8.3], were used to implement phasor sampling routines, capable of sampling at a number of frequencies for every spatial sampling point specified.

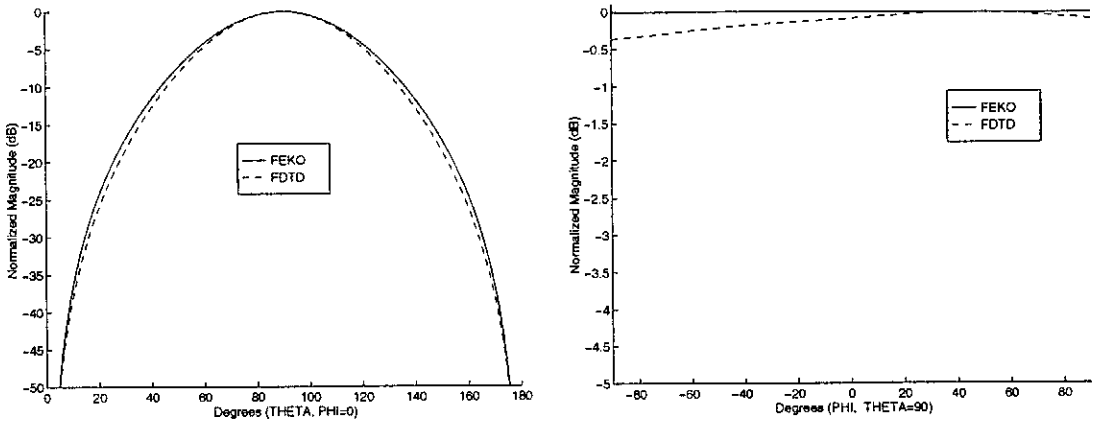
4.5 Near-field Measurement Code Verification

Near-field measurements, from the FDTD, were used in conjunction with the FEKO AP-card to generate far-field patterns. These patterns were then compared to far-field data, generated with FEKO. The results in this section were generated with a resistively loaded dipole, obtained from [22], as structure of interest.

4.5.1 Comments on Simulation Results

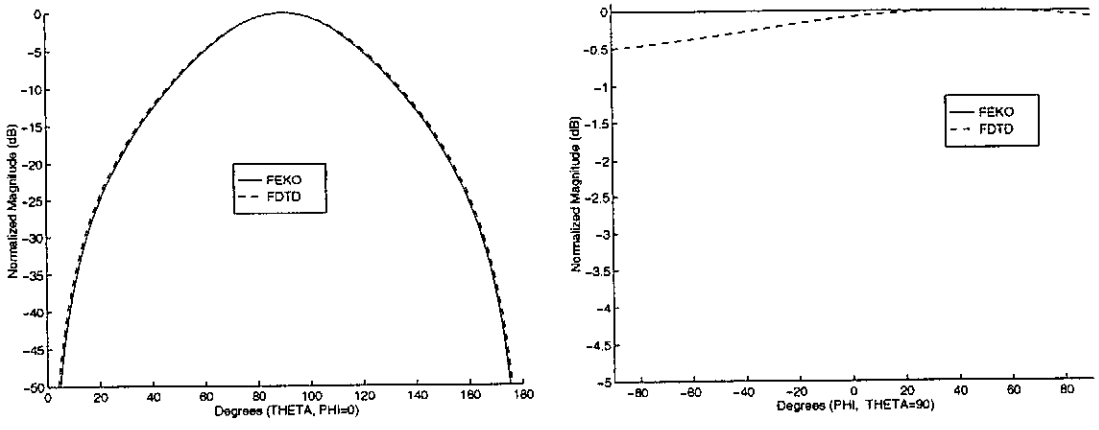
The simulations that were done to produce figures 4.3 through 4.6, were done across the frequency band for which the antenna was designed, [22], to test the near-field sampling formulation for different radiation patterns.

The results of these figures, which were produced with the FDTD sampled near-fields and the FEKO AP-card, show a very good comparison to the FEKO MoM predictions for the far-fields. The phi directivity graphs show less than 0.8dB difference between the MoM and FDTD based simulation results in all cases, while the theta directivity graphs also match to a high degree of accuracy, with some deviation visible in the 1.6GHz and 1.8GHz graphs. The deviation is, however, within reasonable limits of agreement, with both trends and magnitudes comparing favorably.



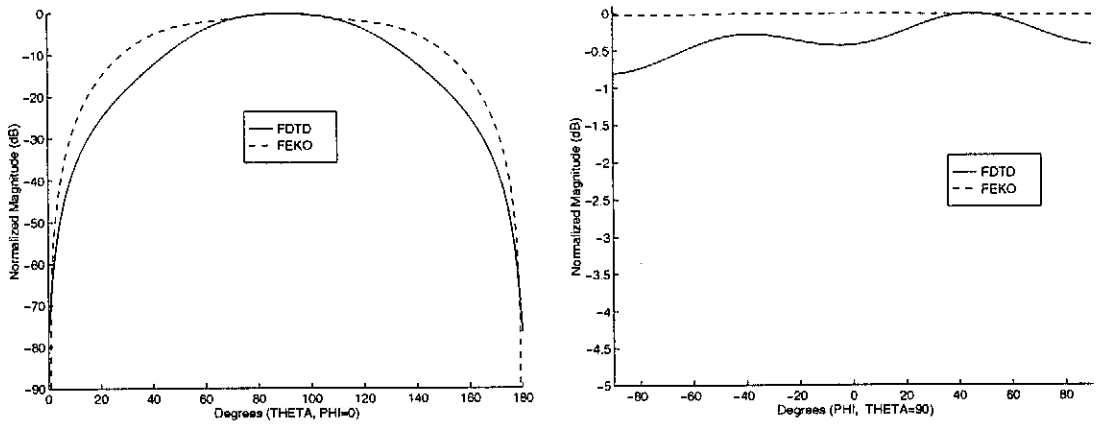
(a) Theta (b) Phi

Figure 4.3: 800MHz Far-field directivity comparison



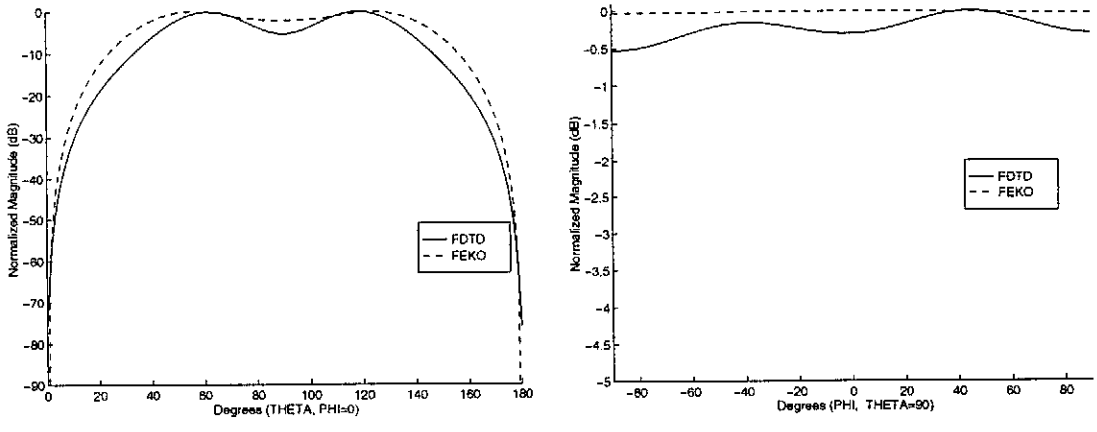
(a) Theta (b) Phi

Figure 4.4: 1.1GHz Far-field directivity comparison



(a) Theta (b) Phi

Figure 4.5: 1.6GHz Far-field directivity comparison



(a) Theta (b) Phi

Figure 4.6: 1.8GHz Far-field directivity comparison

Chapter 5

FDTD Modelling of Dispersive Media

5.1 Introduction to the FDTD and Dispersive Media

5.1.1 Methods of Modelling Dispersive Media in the FDTD

In the modelling of dispersive media in the FDTD, two main methods exist for setting up the numerical model:

- The Recursive Convolution method, and
- The Auxiliary Differential Equation method.

The **Recursive Convolution** method's main feature is that it can model an arbitrary number of Debye and Lorentzian relaxations, whereas the **Auxiliary Differential Equation** method permits modelling of nonlinearities and dispersive nonlinearities in addition to linear dispersions. The Auxiliary Differential Equation method can also model an arbitrary number of Debye and Lorentzian relaxations but does, however, pose increased complexity in modelling as opposed to the Recursive Convolution method.

5.1.2 Relevant Dispersive Media Permittivity Models

The implementation of dispersive media focused on the simulation of water as a dispersive medium. Water is a linear, isotropic, dispersive medium which is adequately modelled by the *Debye first order* formulation for the relative permittivity of a dispersive medium. The first order Debye expression for relative permittivity, as a function of frequency, may be written as follows:

$$\epsilon_r(\omega) = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + j\omega t_0} \tag{5.1}$$

where:

- ϵ_∞ is the relative permittivity of the medium where $\omega = \infty$.

- ϵ_s is the relative permittivity of the medium where $\omega = 0$.
- t_0 is the Debye relaxation time constant for the medium.

In distilled water these constants are, [28]:

- $\epsilon_\infty = 1.8$
- $\epsilon_s = 81$
- $t_0 = 9.4 \times 10^{-12}$

5.1.3 FDTD Dispersive Media Discretization Considerations

In setting up a FDTD spatial grid and time discretization it is important to remember the following rules.

The Courant condition for stability, [5, paragraph 4.6]:

$$\triangle t \leq \frac{1}{v\sqrt{\frac{1}{(\triangle x)^2} + \frac{1}{(\triangle y)^2} + \frac{1}{(\triangle z)^2}}} \tag{5.2}$$

Spatial cell size should be less than the wavelength of the highest frequency of interest. A rule of thumb for the specification of spatial cell size ([6, paragraph 3.2]) may be stated as:

$$\triangle s \leq \frac{\lambda_{min}}{10} \tag{5.3}$$

It is important to remember that these formulae require the speed of light *in the medium* for accurate calculations. This implies that the timesteps that satisfy the Courant condition in freespace should be made smaller by a factor of $\sqrt{\epsilon_r}$ of when working in a dispersive medium. The appropriate choice of ϵ_r should be the highest value of ϵ_r that will be encountered in the simulation. The rule of thumb for the specification of the spatial cell size should then also be divided by $\sqrt{\epsilon_r}$ when in a dispersive medium.

5.2 One Dimensional MATLAB Dispersive Media Investigation

A one dimensional investigation was undertaken to come to terms with the different formulations that describe dispersive media in the FDTD. The goal of these experiments was to write code that would utilize both the Recursive Convolution and the Auxiliary Differential Equation methods to calculate the reflection coefficient of water.

The Recursive Convolution formulation was taken from [6, pp. 125-131], with the derivations being repeated to correct a typographical error[1] in the text. The update equations

[1] $\triangle\chi$ in eq. (8.12) [6, p. 127] should be $\triangle x$.

for the Recursive Convolution method are presented in full 3D form in the Appendix, section A.7.

The Auxiliary Differential Equation formulation was taken from [5, pp. 246-248], but is also discussed in [29][2]. As with the Recursive Convolution formulation the derivations were repeated to correct an error[3] in the text. The 1D update equations for the Auxiliary Differential Equation method are presented in the Appendix, section A.6

5.2.1 Simulation Challenges

Simulation Construction: Air-water Boundary

One of the main considerations to take into account is how to connect the two regions. The H-field updates remain the same for media, such as water, where $\mu_r(\omega)$ is a constant, and since the E-field updates rely only on the H-field components one half-space in front and behind of them, it is clear that the mathematical boundary between the two regions should fall on an H-field point, or in other words, on a half-space point. Update equations take care of the connection between the two regions quite easily with this formulation. The location of the true boundary between the two regions is unfortunately not so easy to define, and is still a topic of debate in the FDTD community.

Analytical Solution and Reliable Pulse-widths

The reflection coefficient for an air/water interface is easy to calculate analytically and such an analytical solution was implemented to evaluate the MATLAB code. The analytical solution of the reflection coefficient shows that the reflection coefficient changes rapidly from approximately 1GHz to 13THz. Although it is theoretically possible to analyze such a wide bandwidth with a single FDTD pulse, it is not numerically practical. Pulses with such a wide bandwidth may be used in a FDTD simulation, but the numbers in the simulation (especially in the "late-time" region of the simulation) become very small and numerical inaccuracies, such as round-off errors, start to corrupt calculated data. These errors require the use of a number of pulses with different frequency content to analyze the system with a high degree of accuracy.

The system was simulated using five different pulses, with the following frequency data being extracted from each run:

- 1GHz to 60GHz, Gaussian pulse.
- 60GHz to 250GHz, Differentiated Gaussian pulse.
- 250GHz to 1THz, Differentiated Gaussian pulse.
- 1THz to 3.5Hz, Differentiated Gaussian pulse.

[2]Ghandi *et al.* also presents interesting theory on the modelling of human tissue as a dispersive medium, which makes the FDTD a good technique to use in Radiation Hazard (RadHaz) investigations.

[3]Both terms on the right-hand side of eq. (9.61) [5, p. 248] should be multiplied by ϵ_0.

- 3.5THz to 12THz, Differentiated Gaussian pulse.

These pulse widths represent the 3dB bandwidth of the pulse. Although the pulses obviously have a much broader frequency spectrum, it was found empirically that data outside the 3dB point started to degrade too much to be trusted with confidence.

Termination and Source Terms

The source for each simulation was a soft source, which was implemented close to the far boundary of the air region. Reflections from the boundary of the air region were attenuated with a first order Mur boundary condition, while the reflections from the boundary of the water region were removed by timegating. The Mur formulation for air is obtainable from [7].

5.2.2 One Dimensional Code Verification

Figure 5.1 shows the comparison between the FDTD solution and the analytical solution for the permittivity of water. The regions where the different pulses were used are also indicated.

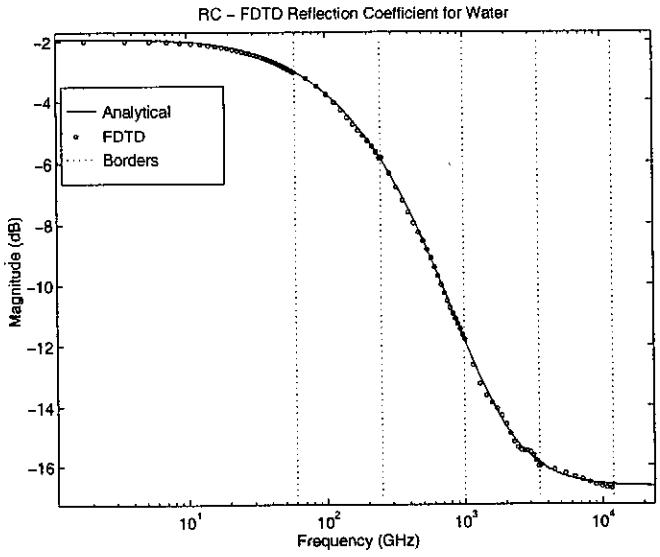


Figure 5.1: Reflection coefficient comparison with frequency regions indicated

5.3 Three Dimensional Dispersive Media FDTD Model

The *Recursive Convolution* method was chosen to model the dispersive medium in Fortran. The main reason for this choice is that the method would be the easiest to implement in the Fortran environment and an added benefit of this model is that the magnetic field updates do not change at all from their freespace form.

The expressions for the electric field updates for a Debye first order dispersive, linear media were derived and the 3D update equations are presented in the Appendix, section A.7.

5.3.1 Absorbing Boundary Conditions

In most FDTD simulations the region of simulation would ideally be unbounded. In numerical reality the region has to be terminated with some sort of numerical function to absorb radiation propagating outward. The scheme that is implemented in this project is the 2nd order Mur ABC, described in [7], [5, pp. 158-160] and [6].

The problem with the standard Mur boundary condition, which makes it inaccurate for use in dispersive media terminations, is that it depends on the speed of propagation of electromagnetic waves in the medium. The literature, cited previously, derives the Mur ABC's for freespace, which could quite easily be adjusted to work in a non-dispersive dielectric medium, but as different frequency components propagate at different speeds in dispersive media, such an adjustment is not a trivial matter when dispersive media has to be terminated.

The main interest of this work, is how electromagnetic radiation would propagate in water at frequencies from 10MHz to 100MHz. Figure 5.2 shows the relative permittivity of water from DC to 5THz. This figure shows quite clearly that water could be approximated as a non-dispersive media with $\epsilon_r = 81$ in the frequency range of interest .

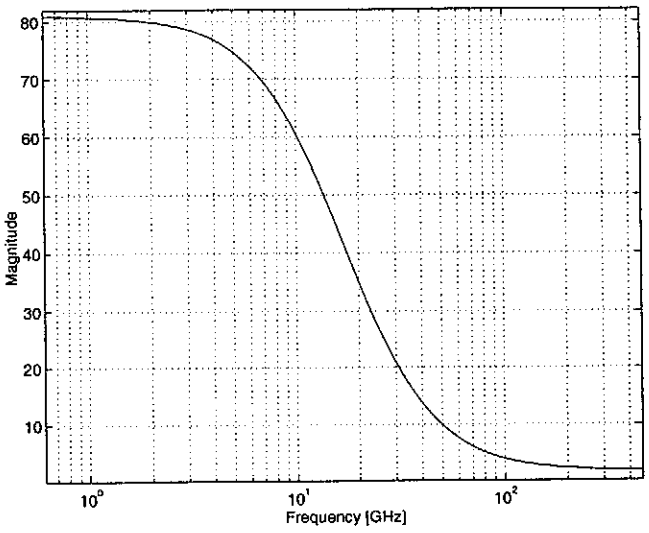


Figure 5.2: Relative permittivity for water

Practical measurements were, however, made and simulated, by Keller, using a scaled model of the antenna of interest ([18, paragraph 3.4]) which operated in the frequency range 200MHz to 2GHz, for reflection coefficient measurements, and from 300MHz to 1.3GHz, for coupling measurements. In these frequency ranges water changes from $\epsilon_r = 81$ to $\epsilon_r = 80.44$ at 2GHz, and $\epsilon_r = 80.76$ at 1.3GHz.

The second order Mur boundary condition equations, for the experiments of this thesis, are therefore adjusted to be correct for a medium with $\epsilon_r = 80.73$, which is the mean value for the three values of interest, and the medium is then assumed to be linear and non-dispersive where the boundary conditions are applied. In all other parts of the code the full RC-formulation update equations are still applied. If more accurate boundary terminations with very small reflections are required, it would be advisable to implement Berenger's PML, [8], which was generalized for application in dispersive media by Uno *et al.* in [13].

5.3.2 Thin Wire Update Equations in Dispersive Media

The impedance loaded dipole of interest to this project also needed to be simulated in a dispersive medium. In order to simulate this antenna in a dispersive medium in the FDTD it is necessary to change the thin wire update equations into a form that would be able to model the dispersive nature of the medium.

The update equations for freespace magnetic fields were derived using Faraday's law and are therefore not influenced in any way by the dispersive medium. Since the electric field components that are normal to the wire propagate without influence from the wire they are simply changed to the normal Ex and Ey update equations (A.44 and A.45) for a dispersive medium.

5.3.3 Lumped Element Update Equations in Dispersive Media

The update equations for lumped elements were derived using Ampére's curl equation, A.3. This equation contains a permittivity dependency in $\overline{D} = \epsilon_0 \epsilon_r(\omega)\overline{E}$, which ordinarily needs to be modelled by a dispersive model. It is, however, important to note that the $\overline{D}$ term refers to the $\overline{E}$-field inside the resistor, which is not dispersive. The equations that describe the update equations for the $\overline{E}$-field inside a lumped elements must therefore remain as they were in freespace.

5.4 Dispersive Media Code Verification

5.4.1 Generating Comparative Data for Test Purposes

Generation of data to compare to the FDTD dispersive media proved to be a problem. No code was available that could handle dispersive media outright, and a method had to be found that could utilize existing numerical tools to perform dispersive media simulations. FEKO presents two options, which are discussed in the Appendix, chapter D.

5.4.2 Simulation Setup

In freespace the diameter of the wire was set to 1.5mm (6mm wide strip), but in a dispersive medium the cell size needs to be made smaller for the high frequency components to propagate properly. The thin wire approximations, by contrast, requires that the cell dimensions be at least twice the radius of the wire that is being simulated. Therefore, the wire needs to be made thinner to compensate for the thin wire approximation limit of half the wire thickness. Figure 5.3 shows that a smaller wire radius of 1.2mm (4.8mm wide strip) has very little effect on the input impedance of an impedance loaded wire antenna in freespace, and is a valid approximation to the true antenna of interest.

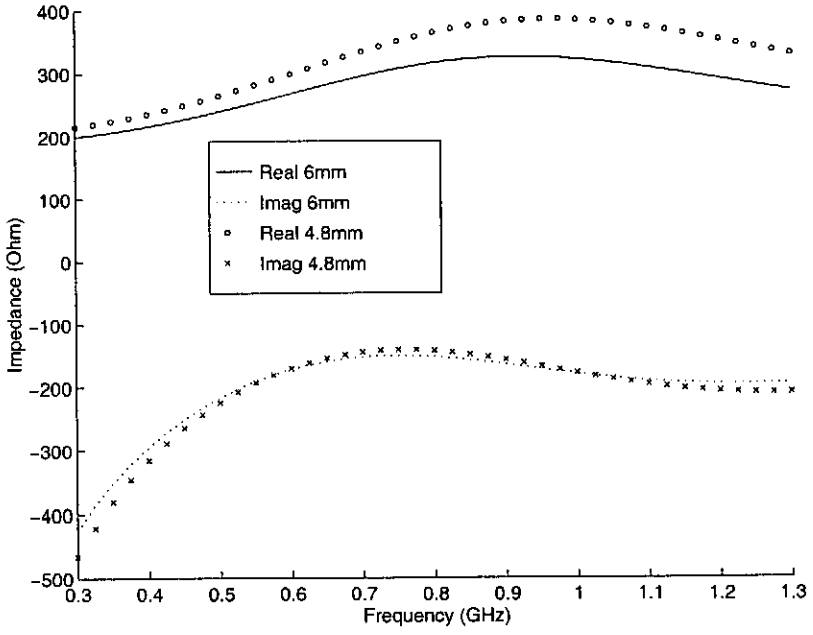


Figure 5.3: Input impedance comparison for a 4.8mm strip to a 6mm strip

5.4.3 Comparative Analysis

The FDTD code was evaluated by comparing the numerically calculated S_{11} of the FEKO and FDTD methods of modelling. Appendix section A.5 explains how the S-parameters may be extracted from a FDTD simulation.

Figure 5.4 compares the input impedance of a 300mm dipole calculated with the FEKO DI-card and the FDTD.

The FDTD curve in figure 5.4 was compiled from the results of two FDTD simulations and shows that the formulation is capable of detecting the resonances in the reflection coefficient for the antenna with accuracy with respect to frequency. It unfortunately does not predict the resonance at 80MHz at the right frequency and shows large deviations in magnitude.

These discrepancies are caused by the fact that the cell size for these simulations were set to 4.16mm, while working in a grid with size limited by the physical resources of the

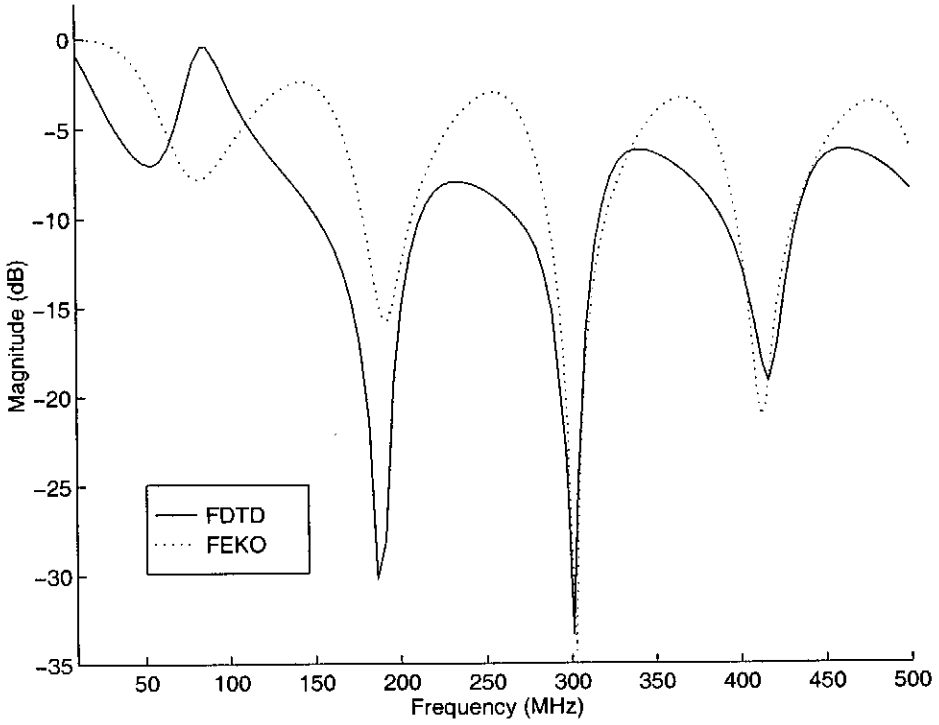


Figure 5.4: S_{11} of a 300mm dipole in water - FDTD 2mm cells (Composite)

available computer, which places the boundary condition for the simulation very close to the antenna at low frequencies. The reflections from the boundary interferes with the antenna and causes the detuning of the resulting input impedance and the deviation in the accuracy of the magnitude of the FDTD simulations.

The classic engineering tradeoff between accuracy in modelling and available computational resources is thus encountered here. The way to solve this problem would be to either:

- Enlarge the problem space at a small discretization until the boundary is far enough from the antenna, or
- Implement a better boundary condition, possibly a Berenger PML for dispersive media as formulated by [13], to suppress the reflections from the boundary in a smaller problem space to a greater degree.

The first option was taken and the simulations were moved from the original Pentium III 533MHz PC, with 128Mb of RAM, to a Silicon Graphics Octane workstation, with two 270MHz MIPS R12000 processors and 1280Mb of RAM.

The simulations that were done on the Pentium III PC only placed the boundary at approximately 12% of a wavelength at 50MHz, and required 18.94Mb of RAM to run. The move to the SGi workstation enabled the use of much larger matrices, placing the boundary at approximately 50% of a wavelength at 50MHz, and required approximately 285Mb of RAM to run.

Figure 5.5 presents the results that were obtained using the SGi workstation, and shows a clear improvement over the results from the Pentium III PC. The magnitudes of the FDTD and FEKO simulations match extremely well, with slight frequency deviations at higher frequencies.

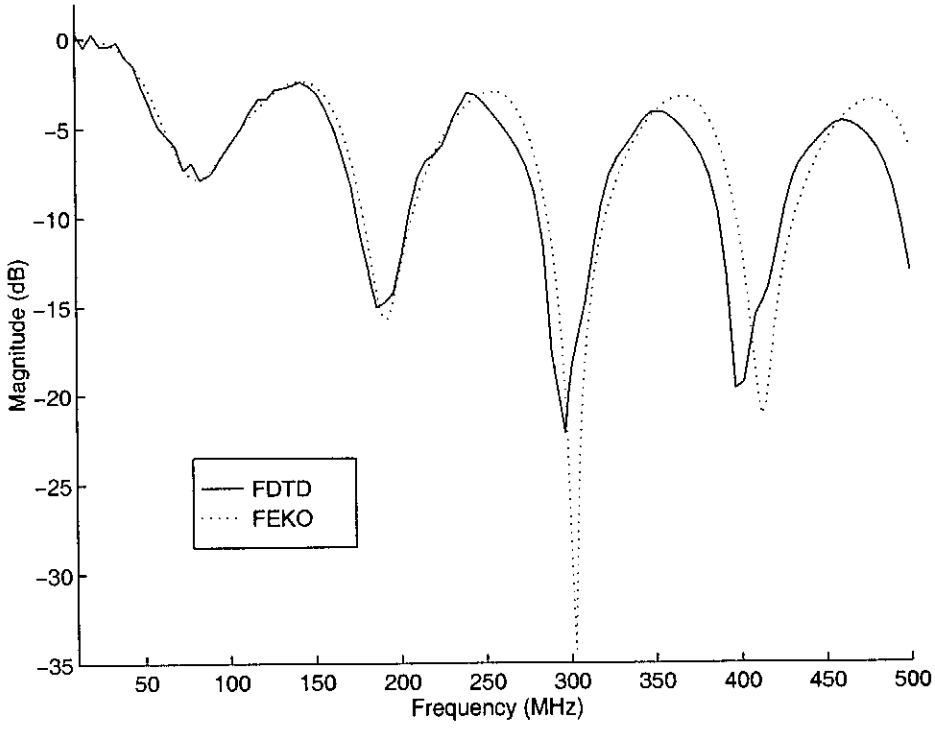


Figure 5.5: S_{11} of a 300mm dipole in water - FDTD 4mm cells (Large problem)

As the Mur boundary condition formulation rests on the assumption that the incident wave is a plane wave, the boundary should ideally be placed in the far-field of the simulation. This insight explains why the simulations improved when the boundary could be moved to 50% of a wavelength.

Chapter 6

Conclusions

6.1 Achievements

The aim of this project was to master the fundamentals of the FDTD and to create the basis of a simulation tool for the complex radiation and propagation environments of the *Deepmine* project. In pursuing these goals the following features of the FDTD were studied and implemented in the code:

- Thin wire antennas, with subcellular radii, were implemented using a fine geometric model, which is also the basis for the modelling of a number of irregular structures in regular FDTD grids.
- Lumped element models for resistors and capacitors. These models were used in conjunction with thin wires to create a FDTD model for impedance loaded dipoles. The theory behind the FDTD formulation of resistors and capacitors is also applicable to the formulation of a number of other discrete components, such as inductors, diodes and even transistors, although the abilities of the FDTD to handle these non-linear elements were not used in this project.
- Near-field sampling of specific frequency components in phasor format, for the estimation of far-field radiation patterns. The sampling of phasors also required the implementation of a DFT that could be performed "on-the-fly".
- Dispersive media formulations were studied and evaluated to select the most appropriate model for implementation in this code. The Recursive Convolution and Auxiliary Differential Equation formulations were mastered, with the Recursive Convolution formulation being implemented in 3D. The thin wire models, that previously only existed for freespace, were also implemented in dispersive media with success.
- OpenGL was investigated with Fortran GUI visualization as a goal, and the study presented in the Appendix, chapter E. It was shown that the coding tools, that are required to develop a GUI within Fortran, exists and can be used with ease and efficiency.

6.2 Future Development Opportunities

Although the work that has been done represents a solid foundation for the development of a FDTD code to simulate borehole radar, there will always be room for improvement and further development. Some of the future development opportunities include:

- The implementation of stratified media. The FDTD is extremely well suited to the simulation of inhomogeneous environments with permittivity changing rapidly or dispersively. Each cell in a simulation could quite easily be assigned a specific permittivity or permeability, without adding to the complexity of the simulation. The implementation of stratified media is therefore almost entirely a coding and meshing problem, and not so much a modelling problem. Future development could update the code to be able to handle inhomogeneous media by modifying the update equations and boundary conditions accordingly. Experience gathered from this exercise would mainly be in coding and not so much in modelling.

- The implementation of a better boundary condition. The Mur ABC, although sufficient for the work presented in this thesis, should be replaced with a better boundary condition. Implementation of a Berenger PML would be the obvious suggestion for a new boundary condition, and would afford the opportunity to study the intricacies of the PML, while increasing the dynamic range of the code appreciably. Both modelling and coding experience would be gained from such an implementation.

- The implementation of local near-field to far-field transformations. Green's functions for freespace and dielectric media could be investigated and implemented, utilizing the phasor fields that the code already provides, to implement near-field to far-field transformations. Experience in modelling and coding would be gathered from this exercise.

- The implementation of a GUI for the code. At present the code uses a number of text files and hardcoded sources of data to describe the problem that is being simulated, and a GUI would greatly enhance the functionality of the code. An appropriate GUI should include features for the visualization of the problem space and the input of features into this problem space. This exercise would only provide coding experience.

- Parallelization of the code. The FDTD lends itself extremely well to parallelization, as each field point that is being updated is determined only by values that are constant at that timestep. When the E-field matrix or the H-field matrix is updated, it could be split up into a number of blocks, which could each be updated separately. Davidson and Ziolkowski implemented such a FDTD code, in [30], and derived expressions that describe the efficiency to be gained from parallelization of a FDTD code. Experience would be gathered in coding for a multiprocessor environment.

6.3 Concluding Remarks

This thesis shows that the FDTD method is applicable to the simulation of borehole radars and that it is capable of handling the complex geophysical phenomenon associated with the mining environment. It successfully developed a code to form the basis of a simulation environment for the *Deepmine* project, and in so doing achieves the goals that were set for the project.

Appendix A

Mathematical Derivations, Proofs and Laws

A.1 Maxwell's Equations in Three Dimensions

Maxwell's equations are used in either their integral of differential forms to derive every FDTD update equation. The Laws of interest are given below.

Faraday's Law:

$$\frac{d\overline{B}}{dt} = -\nabla \times \overline{E} - \overline{J}_m \tag{A.1}$$

$$\frac{d}{dt}\iint_S \overline{B}.d\widehat{S} = -\oint_C \overline{E}.d\widehat{l} - \iint_S \overline{J}_m.d\widehat{S} \tag{A.2}$$

Ampére's Law:

$$\frac{d\overline{D}}{dt} = \nabla \times \overline{H} - \overline{J}_e \tag{A.3}$$

$$\frac{d}{dt}\iint_S \overline{D}.d\widehat{S} = \oint_C \overline{H}.d\widehat{l} - \iint_S \overline{J}_c.d\widehat{S} \tag{A.4}$$

Gauss's Law for the Electric Fields:

$$\nabla.\overline{D} = 0 \tag{A.5}$$

$$\oiint_S \overline{D}.d\widehat{S} = 0 \tag{A.6}$$

Gauss's Law for Magnetic Fields:

$$\nabla.\overline{B} = 0 \tag{A.7}$$

$$\oiint_S \overline{B}.d\widehat{S} = 0 \tag{A.8}$$

where:

- $\overline{E}$ is the electric field vector in Volts per meter,
- $\overline{D}$ is the electric flux density vector in Coulombs per square meter,
- $\overline{H}$ is the magnetic field vector in Ampéres per meter,
- $\overline{B}$ is the magnetic flux density vector in Webers per square meter,
- $\overline{J}_e$ is the electric conduction current density in Ampéres per square meter,
- $\overline{J}_m$ is the equivalent magnetic conduction current in Volts per square meter,
- S is an arbitrary surface with the unit normal vector $d\widehat{S}$, and
- C is the contour that bounds S with the unit path length vector $\widehat{dl}$

In linear, isotropic nondispersive materials (i.e. materials having field-independent, direction-independent and frequency-independent electric and magnetic properties), $\overline{B}$ is related to $\overline{H}$ and $\overline{D}$ is related to $\overline{E}$ in the following relations:

$$\overline{B} = \mu\overline{H} \tag{A.9}$$
$$\overline{D} = \epsilon\overline{E} \tag{A.10}$$

where:

- μ is the magnetic permeability in Henrys per meter, and
- ϵ is the electric permittivity in Farads per meter.

Magnetic loss currents may be defined as:

$$\overline{J}_m = \rho'\overline{H} \tag{A.11}$$

Electric loss currents may be defined as:

$$\overline{J}_e = \sigma\overline{E} \tag{A.12}$$

A.2 Analytical Solution for the Reflection Coefficient of Water

The derivations that were used in obtaining the analytical solution for the reflection coefficient of water, were all made under the assumption of normal incidence, [24]. The formulae required in this work are:

$$k = \omega.\sqrt{\mu\epsilon} \tag{A.13}$$

$$\eta = \sqrt{\frac{\mu}{\epsilon}} \tag{A.14}$$

$$\rho = \frac{\eta_2 - \eta 1}{\eta_2 + \eta 1} \tag{A.15}$$

where:

- ρ is the reflection coefficient when a plane wave from medium one reflects from the interface with medium two.
- ϵ should be written as $\epsilon_0\epsilon_r$ for medium two.
- η_1 is the wave impedance for the first medium (air).
- η_2 is the wave impedance for the second medium (water).
- The incident plane wave originates in medium one (air) and reflects from the boundary with medium two (water).

The relative permittivity for water, required in equation A.14, was plotted in figure 5.2.

A.3 Fine Geometric Model - Derivation of H_y for a Thin Wire Update Equation

In deriving the update equations for a thin wire under the laws of the Fine Geometric Model, Faraday's integral law A.2 is discretized and solved for H_y by analyzing the two sides of the equation in separate steps. Figure A.1[1] is helpful in the interpretation of the integrals.

Faraday's law must be put into a different form before the derivation is done. Using the identity $\overline{B} = \mu_0\overline{H}$, and figure A.1, Faraday's law now becomes:

$$\frac{d}{dt}\iint_{S_2} \mu_0 H_y dz dx = \oint_{C_2} \overline{E}.dl_2 \tag{A.16}$$

The lefthand side, of equation A.16, may now be written as:

$$\frac{d}{dt}\int_0^{\Delta}\int_{z_0-\frac{\Delta}{2}}^{z_0+\frac{\Delta}{2}} \left(\mu_0 H_y|_{\frac{\Delta}{2},z_0}.\frac{\frac{\Delta}{2}}{x}\left[1 + c_1(z - z_0)\right]\right) dz dx \tag{A.17}$$

Once the constants have been removed from the integrand, a simple geometric integral is left. This integral does, however, reduce to an analytic form a great deal simpler if the

[1] Originally sourced from [31]

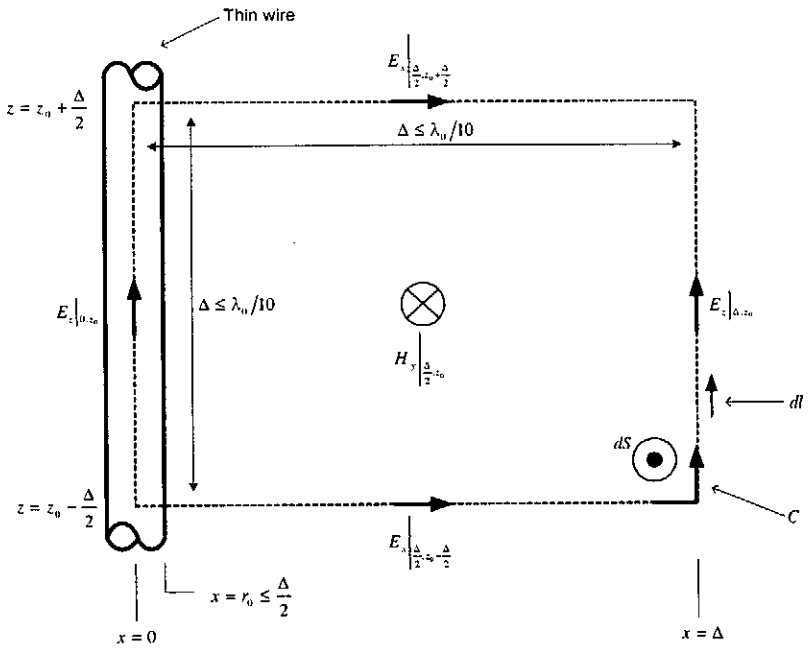


Figure A.1: Faraday's Law contour path for the thin wire

integration with respect to z is done first. The left side of equation A.16 then simplifies to:

$$\frac{d}{dt}\iint_{S_2}\mu_0 H_y dz dx = \mu_0 . \frac{\Delta}{2} . \ln\left(\frac{\Delta}{r_0}\right) . \frac{\left(H_y|^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k} - H_y|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k}\right)}{\Delta t} . \Delta \tag{A.18}$$

The righthand side, of equation A.16, may now be written as:

$$\begin{aligned}
\oint_{C_2} \overline{E}.\widehat{l}_2 &= \int_{z_0-\frac{\Delta}{2}}^{z_0+\frac{\Delta}{2}} \left[E_z(\Delta, z) - E_z(0, z)\right] dz \\
&\quad + \int_{r_0}^{\Delta} \left[E_x(x, z_0 - \frac{\Delta}{2}) - E_x(x, z_0 + \frac{\Delta}{2})\right] dx \qquad \text{(A.19)} \\
&= \int_{z_0-\frac{\Delta}{2}}^{z_0+\frac{\Delta}{2}} \left(E_z|_{\Delta,z} - E_z|_{0,z}\right) \left[1 + c_2(z - z_0)\right] dz \\
&\quad + \int_{r_0}^{\Delta} \left(E_x(|_{\frac{\Delta}{2},z_0-\frac{\Delta}{2}} - E_x|_{\frac{\Delta}{2},z_0+\frac{\Delta}{2}}\right) \left[\frac{\frac{\Delta}{2}}{x}\right] dx \qquad \text{(A.20)} \\
&= \left(E_z|^n_{i+1,j,k} - E_z|^n_{i,j,k}\right) . \Delta \\
&\quad + \left(E_x|_{i+\frac{1}{2},j,k-\frac{1}{2}} - E_x|_{i+\frac{1}{2},j,k+\frac{1}{2}}\right) . \frac{\Delta}{2} . \ln\left(\frac{\Delta}{r_0}\right) \qquad \text{(A.21)}
\end{aligned}$$

Equating equations A.18 and A.21 results in the required update equation.

A.3.1 Thin Wire Update Equations in Three Dimensions

These update equations use Taflove's notation, [5, paragraph 3.6.2], and further define:

- r_0 as the radius of the wire.
- Δ as the spatial discretization of the FDTD grid.

In the cells containing the wire, the update equations are:

$$H_y|_{i,j,k}^{n+\frac{1}{2}} = H_y|_{i,j,k}^{n-\frac{1}{2}} + \Delta t \frac{(E_x|_{i,j,k}^{n} - E_x|_{i,j,k+1}^{n})\frac{1}{2}\ln(\frac{\Delta}{r_0})}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} + \frac{E_z|_{i+1,j,k}^{n}}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} \quad \text{(A.22)}$$

$$H_x|_{i,j,k}^{n+\frac{1}{2}} = H_x|_{i,j,k}^{n-\frac{1}{2}} + \Delta t \frac{(E_y|_{i,j,k+1}^{n} - E_y|_{i,j,k}^{n})\frac{1}{2}\ln(\frac{\Delta}{r_0})}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} - \frac{E_z|_{i,j+1,k}^{n}}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} \quad \text{(A.23)}$$

$$E_z|_{i,j,k} = 0 \quad \text{(A.24)}$$

These equations are valid for z-directed wires, but x and y directed wires may also be implemented by inspection, with appropriate changes being made to the relevant field component update equation.

A.3.2 Deltagap Voltage Source Update Equations

The update equations for the deltagap fed section of wire are:

$$H_y|_{i,j,k}^{n+\frac{1}{2}} = H_y|_{i,j,k}^{n-\frac{1}{2}} + \Delta t \frac{(E_x|_{i,j,k}^{n} - E_x|_{i,j,k+1}^{n})\frac{1}{2}\ln(\frac{\Delta}{r_0})}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} + \frac{(E_z|_{i+1,j,k}^{n} - E_z|_{i,j,k}^{n})}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} \quad \text{(A.25)}$$

$$H_x|_{i,j,k}^{n+\frac{1}{2}} = H_x|_{i,j,k}^{n-\frac{1}{2}} + \Delta t \frac{(E_y|_{i,j,k+1}^{n} - E_y|_{i,j,k}^{n})\frac{1}{2}\ln(\frac{\Delta}{r_0})}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} - \frac{(E_z|_{i,j+1,k}^{n} + E_z|_{i,j,k}^{n})}{\mu_0 \ln(\frac{\Delta}{r_0})\frac{\Delta}{2}} \quad \text{(A.26)}$$

where $E_z|_{i,j,k}^{n}$ represents the arbitrarily specified source term that is fed into the gap.

A.3.3 Deltagap Resistor Update Equations

The update equations for a section of wire with a resistor in the center are:

$$E_z|_{i,j,k}^{n+1} = \left(\frac{1 - \frac{\Delta t \Delta z}{\epsilon_0 2R\Delta x\Delta y}}{1 + \frac{\Delta t \Delta z}{\epsilon_0 2R\Delta x\Delta y}}\right) E_z|_{i,j,k}^{n} + \left(\frac{\frac{\Delta t}{\epsilon_0}}{1 + \frac{\Delta t \Delta z}{\epsilon_0 2R\Delta x\Delta y}}\right) \nabla \times H|_{i,j,k}^{n+\frac{1}{2}} \quad \text{(A.27)}$$

where

$$\nabla \times H|_{i,j,k}^{n+\frac{1}{2}} = \frac{(H_y|_{i,j,k}^{n+\frac{1}{2}} - H_y|_{i-1,j,k}^{n+\frac{1}{2}})}{\Delta x} - \frac{(H_x|_{i,j,k}^{n+\frac{1}{2}} - H_x|_{i,j-1,k}^{n+\frac{1}{2}})}{\Delta y} \quad \text{(A.28)}$$

and R is the resistance value of the resistor in Ohm.

A.3.4 Deltagap Capacitor Update Equations

The update equations for a section of wire with a capacitor in the center are:

$$E_z|_{i,j,k}^{n+1} = E_z|_{i,j,k}^{n} + \left(\frac{\frac{\triangle t}{\epsilon_0}}{1 + \frac{C \triangle z}{\epsilon_0 \triangle x \triangle y}} \right) \nabla \times H|_{i,j,k}^{n+\frac{1}{2}} \tag{A.29}$$

where C is the capacitance value of the capacitor in Farad and $\nabla \times H|_{i,j,k}^{n+\frac{1}{2}}$ has the same meaning as in equation A.28.

A.4 Input Impedance Equations and Derivations

The expression used in calculating the input impedance of a structure is:

$$Z_0(\omega, x_i) = \frac{\mathcal{F}[V(t, x_i)].e^{-j\omega\frac{\triangle t}{2}}}{\mathcal{F}\left[\sqrt{I(t, x_{i+\frac{1}{2}}).I(t, x_{i-\frac{1}{2}})}\right]} \tag{A.30}$$

where $\mathcal{F}$ signifies a Fourier transform.

In using this equation the following laws are used:

$$V(t, x_i) = \int_S \overline{E}.d\widehat{S} \tag{A.31}$$

$$I(t, x_i) = \oint_C \overline{E}.d\widehat{l} \tag{A.32}$$

Using figure A.2 it is easily found that:

$$I(t, x_i) = (H_y|_{i-1,j,k} - H_y|_{i,j,k}).\triangle y + (H_x|_{i,j,k} - H_x|_{i,j-1,k}).\triangle x \tag{A.33}$$

and $V(t, x_i)$ is the magnitude of the feed pulse in time.

A.5 S-parameter Measurements in the FDTD

This is an exercise that needs to be handled with some care. The equation for the S-parameters of any port in a circuit may be written as follows [5, p. 434]:

$$S_{mn}(\omega, x_m, x_n) = \frac{V_m(\omega, x_m)}{V_n(\omega, x_n)} \cdot \sqrt{\frac{Z_{0,n}(\omega)}{Z_{0,m}(\omega)}} \tag{A.34}$$

where:

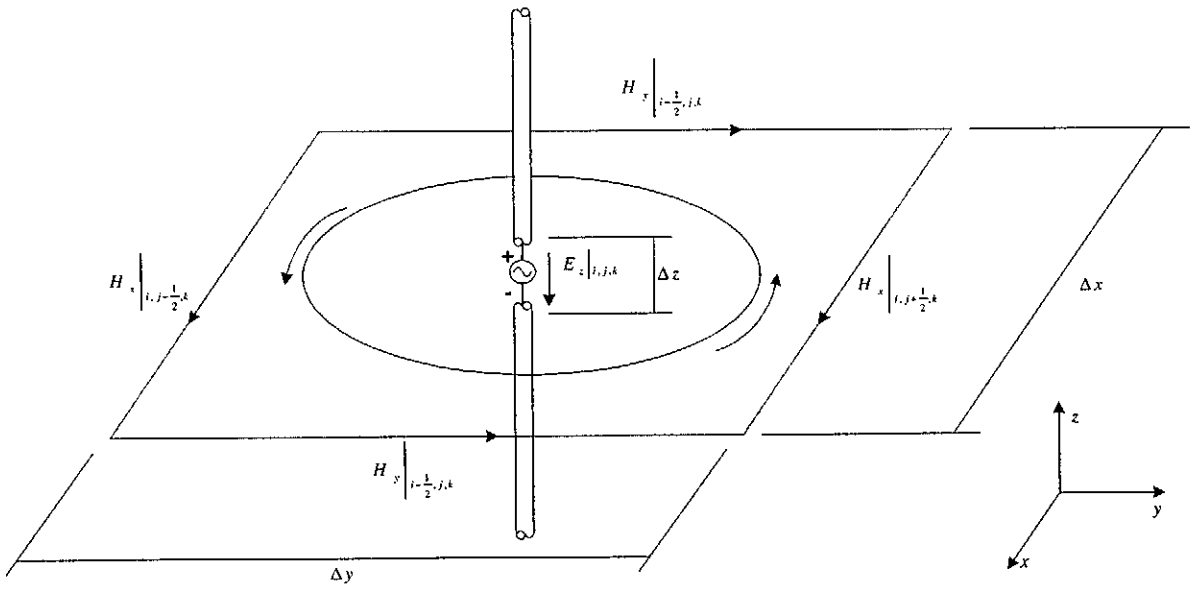


Figure A.2: Input impedance derivation field, voltage and current definitions

- $V_m(\omega, x_m)$ is the voltage at port m, at observation plane x_m, and $V_n(\omega, x_n)$ is the voltage at port n at observation plane x_n, and
- $Z_{0,n}(\omega)$ and $Z_{0,m}(\omega)$ are the characteristic impedances of the lines connected to these ports.

This would mean that to measure S_{11} the incident and reflected pulses at port 1 have to be known, as described in the derivation of S-parameters in [32]. It also means that the characteristic impedance of the line attached to the port may remain unknown, because it cancels itself in the S-parameter equation.

Another way of finding the S-parameters of a circuit is to measure Z-parameters and to convert these Z-parameters to S-parameters, using the relations given in [33]. For the calculation of S_{11} this may be written as:

$$S_{11} = \frac{\frac{Z_{in}}{Z_0} - 1}{\frac{Z_{in}}{Z_0} + 1} \tag{A.35}$$

where Z_0 signifies the characteristic impedance of the transmission line attached to the antenna.

A.6 Auxiliary Differential Equation One Dimensional Update Equations for the First Order Debye Model

The 1D update equations for the Auxiliary Differential Equation method may be written as follows:

$$D_x|_{i,j,k}^{n+1} = D_x|_{i,j,k}^{n} - \frac{\Delta t}{\Delta z}.\left(H_y|_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}} - H_y|_{i,j,k-\frac{1}{2}}^{n+\frac{1}{2}}\right) \quad \text{(A.36)}$$

$$E_x|_{i,j,k}^{n+1} = \frac{2\epsilon_0\epsilon_\infty t_0 - \epsilon_0\epsilon_\infty\Delta t}{2\epsilon_0\epsilon_\infty t_0 + \epsilon_0\epsilon_\infty\Delta t}.E_x|_{i,j,k}^{n} + \frac{\Delta t + 2t_0}{2\epsilon_0\epsilon_\infty t_0 + \epsilon_0\epsilon_\infty\Delta t}.D_x|_{i,j,k}^{n+1} + \frac{\Delta t - 2t_0}{2\epsilon_0\epsilon_\infty t_0 + \epsilon_0\epsilon_\infty\Delta t}.D_x|_{i,j,k}^{n} \quad \text{(A.37)}$$

A.7 Derivation of the Recursive Convolution Update Equations in Three Dimensions for the First Order Debye Model

The derivation of the update equations for a dispersive medium, using the Recursive Convolution method is related in [6, pp.125-129] for a simplified 1D derivation. The derivation of the 3D equations would, however, require that equation (8.12) in [6] be available in a "vector" format, to enable the 3D update equations to be derived from it. The required derivations were done and the resulting equations are:

$$\overline{E}^{n+1} = \frac{\epsilon_\infty}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}}.\overline{E}^{n} + \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}}.\psi^{n} + \frac{\Delta t}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}}.\left(\nabla \times \overline{H}^{n+\frac{1}{2}}\right) \quad \text{(A.38)}$$

where,

$$\chi^0 = (\epsilon_s - \epsilon_\infty)\left[1 - e^{-\frac{\Delta t}{t_0}}\right] \quad \text{(A.39)}$$

$$\Delta\chi^0 = (\epsilon_s - \epsilon_\infty)\left[1 - e^{-\frac{\Delta t}{t_0}}\right]^2 \quad \text{(A.40)}$$

$$\psi^n = \overline{E}^{n}.\Delta\chi^0 + e^{-\frac{\Delta t}{t_0}}.\psi^{n-1} \quad \text{(A.41)}$$

$$\psi^{-1} = 0 \quad \text{(A.42)}$$

$$\nabla \times \overline{H}^{n+\frac{1}{2}} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{d}{dx} & \frac{d}{dy} & \frac{d}{dz} \\ H_x & H_y & H_z \end{vmatrix} = \hat{i}\left(\frac{dH_z}{dy} - \frac{dH_y}{dz}\right) + \hat{j}\left(\frac{dH_x}{dz} - \frac{dH_z}{dx}\right) + \hat{k}\left(\frac{dH_y}{dx} - \frac{dH_x}{dy}\right) \quad \text{(A.43)}$$

The constants that define the medium (ϵ_∞, ϵ_0, t_0 and σ) were given in section 5.1.2.

Update Equations for Propagation in a Homogeneous Dispersive Medium in Three Dimensions

The 3D update equations for the Recursive Convolution method may be written as follows:

$$\begin{aligned} E_x|_{i,j,k}^{n+1} &= \frac{\epsilon_\infty}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \cdot E_x|_{i,j,k}^{n} \\ &+ \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \cdot \sum_{m=0}^{n-1} E_x|_{i,j,k}^{n-m} \Delta\chi^m \\ &+ \frac{\Delta t}{\epsilon_0 \Delta y} \cdot \left(H_z|_{i,j,k}^{n+\frac{1}{2}} - H_z|_{i,j-1,k}^{n+\frac{1}{2}} \right) \cdot \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \\ &- \frac{\Delta t}{\epsilon_0 \Delta z} \cdot \left(H_y|_{i,j,k}^{n+\frac{1}{2}} - H_y|_{i,j,k-1}^{n+\frac{1}{2}} \right) \cdot \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \end{aligned} \tag{A.44}$$

$$\begin{aligned} E_y|_{i,j,k}^{n+1} &= \frac{\epsilon_\infty}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \cdot E_y|_{i,j,k}^{n} \\ &+ \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \cdot \sum_{m=0}^{n-1} E_y|_{i,j,k}^{n-m} \Delta\chi^m \\ &+ \frac{\Delta t}{\epsilon_0 \Delta z} \cdot \left(H_x|_{i,j,k}^{n+\frac{1}{2}} - H_x|_{i,j,k-1}^{n+\frac{1}{2}} \right) \cdot \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \\ &- \frac{\Delta t}{\epsilon_0 \Delta x} \cdot \left(H_z|_{i,j,k}^{n+\frac{1}{2}} - H_z|_{i-1,j,k}^{n+\frac{1}{2}} \right) \cdot \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \end{aligned} \tag{A.45}$$

$$\begin{aligned} E_z|_{i,j,k}^{n+1} &= \frac{\epsilon_\infty}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \cdot E_z|_{i,j,k}^{n} \\ &+ \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \cdot \sum_{m=0}^{n-1} E_z|_{i,j,k}^{n-m} \Delta\chi^m \\ &+ \frac{\Delta t}{\epsilon_0 \Delta x} \cdot \left(H_y|_{i,j,k}^{n+\frac{1}{2}} - H_y|_{i-1,j,k}^{n+\frac{1}{2}} \right) \cdot \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \\ &- \frac{\Delta t}{\epsilon_0 \Delta y} \cdot \left(H_x|_{i,j,k}^{n+\frac{1}{2}} - H_x|_{i,j-1,k}^{n+\frac{1}{2}} \right) \cdot \frac{1}{\epsilon_\infty + \chi^0 + \frac{\sigma\Delta t}{\epsilon_0}} \end{aligned} \tag{A.46}$$

Appendix B

Implementing a Discrete Fourier Transform

The basic DFT equation (DFT analysis from [34, p. 401] and Euler identity from [35, p. 132]) and code fragments from [5, pp. 210-213] were used to implement a DFT that is computed "on-the-fly".

In the form stated in [34] the DFT is a summation that requires knowledge of all the previous values of the sampled signal. In the FDTD environment this is very inconvenient as memory requirements become prohibitive. According to [5] it is conveniently possible to perform the DFT recursively and "on-the-fly". The basic DFT may be written as:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(n) e^{-j\omega_k n \Delta t} \tag{B.1}$$

and from here it is easily proven that the recursive form of this equation may be written as:

$$\begin{aligned} X(\omega_k)|^N &= X(\omega_k)|^{N-1} + x(N-1)e^{-j\omega_k(N-1)\Delta t} && \text{(B.2)} \\ &= X(\omega_k)|^{N-1} + x(N-1)\cos\omega_k(N-1)\Delta t \\ &\quad -jx(N-1)\sin\omega_k(N-1)\Delta t && \text{(B.3)} \end{aligned}$$

A flowchart of the DFT that is used to take frequency samples in the near-field is presented in figure B.1.

Furse, [36], presents evidence that proves the superiority of the DFT over the FFT in the FDTD environment. According to Furse some of the reasons to use the DFT, instead of the FFT, in a FDTD application are:

- The basic summations for the DFT and FFT are the same, and therefore their theoretical accuracy is equivalent.
- In the FDTD environment the DFT is often more computationally efficient than the FFT, and requires a great deal less computing resources.

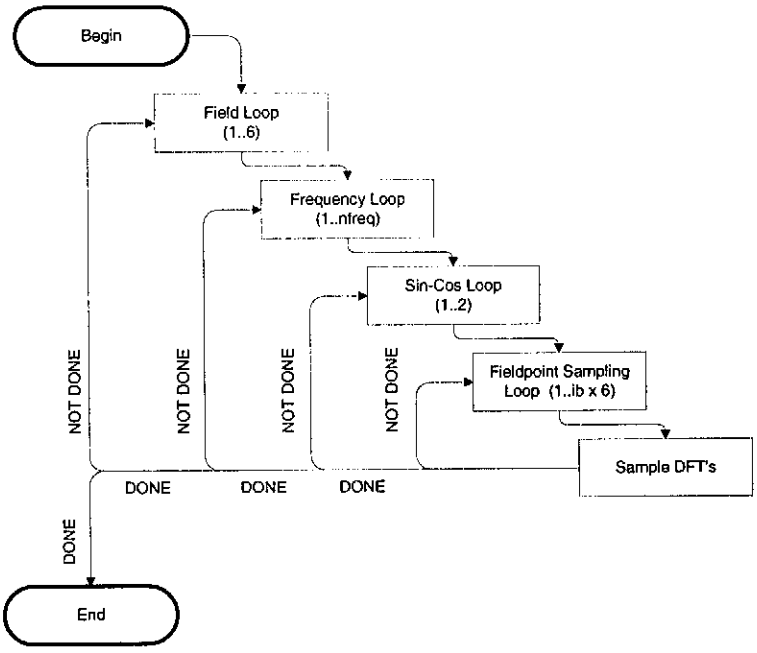


Figure B.1: Flowchart for the DFT sampling algorithm

- The DFT may be computed at specifiable frequencies, where as the FFT is limited by its efficient radix-2 computation.

Appendix C

Empirical Studies into Near-field Sampling

Empirical studies were done to:

- Determine the effect of using only some planes as source for a far-field calculation.
- Determine at what distance from the object of interest (antenna in this case) such measurements should be made to maintain a high level of accuracy.

Results for these studies are shown in figures C.1 and C.2.

Figures C.1.(a) and C.1.(b), show comparative data for far-field patterns, calculated from:

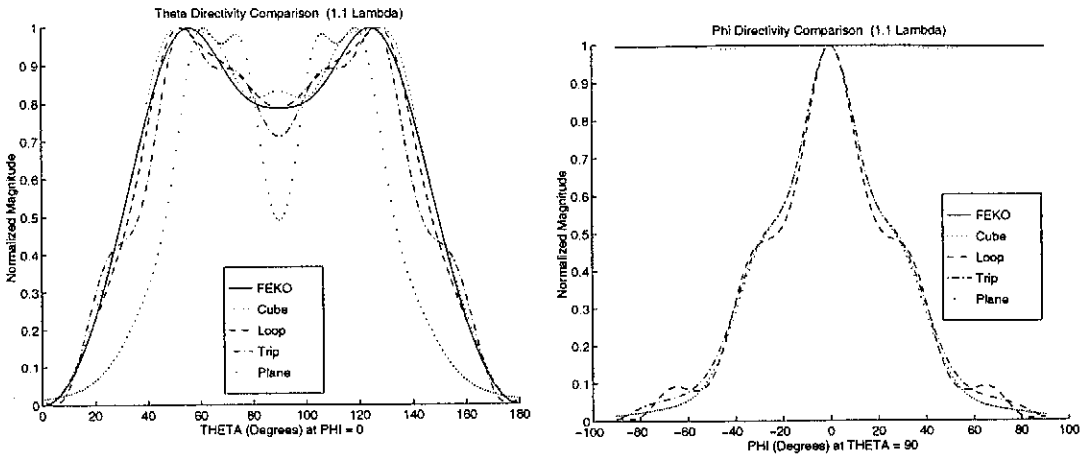
- True FEKO far-field calculations. (No near-fields involved.)
- The near-field data of a cube sampled around the antenna.
- The near-field data of four planes, sampled in a "loop" around the antenna, with center positions at $\phi = 0°$ and θ at 0°, 90°, 180° and 270° in turn.
- The near-field data of three planes, sampled with center positions at $\phi = 0°$ and θ at 0°, 90° and 270° in turn.
- The near-field data of a single plane sampled at $\phi = 0°$ and $\theta = 90°$.

The planes were all sampled at a distance of 1.1λ from the antenna.

Figure C.1 clearly shows that it is advisable to specify an entire cube around the structure of interest as this provides a much better near-field to far-field transformation.

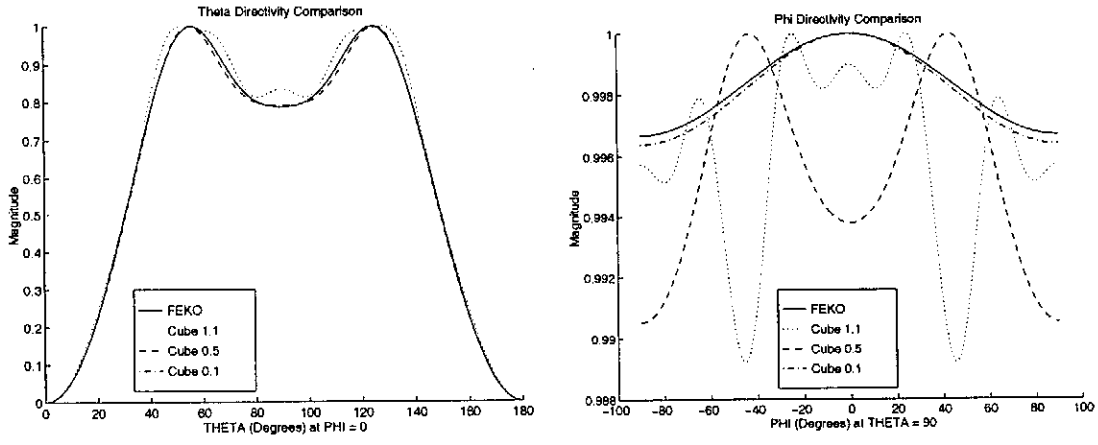
Figures C.2.(a) and C.2.(b) compare the far-field patterns of FEKO to the far-field patterns calculated from the near-field data of a cube. The cube was sampled at different distances from the antenna each time.

Figure C.2 further specifies that the transformation of near-field to far-field is also improved if the near-field is sampled closer to the structure of interest. In the FDTD environment the only limiting factor for near-field sampling would be the physical structure of the spatial grid. It might not be possible to sample at an arbitrary density.



(a) Comparison θ (b) Comparison ϕ

Figure C.1: Sampling configuration comparison



(a) Comparison θ (b) Comparison ϕ

Figure C.2: Sampling distance comparison

Appendix D

FEKO Dispersive Media Modelling

D.1 Introduction to FEKO Dispersive Media Modelling

FEKO presents two possible options for the modelling of dispersive media:

- The dielectrically layered media GF-card or
- The surface current based DI-card .

In both these formulations the specification of a dielectric medium requires the $\tan\delta$ number and the relative permittivity for the medium, which is derived from the Debye first order formula for a dispersive dielectric, as follows:

$$\tan\delta = \frac{\epsilon''}{\epsilon'} \tag{D.1}$$

$$\epsilon' = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1+\omega^2 t_0^2} \tag{D.2}$$

$$\epsilon'' = \frac{-\omega t_0(\epsilon_s - \epsilon_\infty)}{1+\omega^2 t_0^2} \tag{D.3}$$

In an updated version of FEKO a feature is provided that enables the modelling of dispersive media. A FOR-NEXT loop was made available which may be used to implement a frequency loop, which adjusts the relative permittivity of the medium for each frequency point. The FOR-NEXT loop steps through the frequencies of interest and should contain the following statements for the calculation of input impedance or S_{11}:

- A statement that sets the frequency of interest. (FR-card)
- Statements that calculate the relative permittivity.
- A statement that sets the dielectric media properties. (GF-card or DI-card)
- A statement that specifies the voltage excitation.
- A statement that specifies the saving of currents. (OS-card)

D.2 FEKO GF-card Setup

The FEKO GF-card may be used to model dispersive media, keeping the following in mind. Since the GF-card is used with only one medium in mind, only one dielectric layer needs to be specified, but the GF-card requires that two levels be specified, as layer zero is included in all analyses but is not specifiable. Level one should therefore be assigned as the dispersive media layer. The following points should be noted in the setup of the GF-card analysis:

- A planar multilayer Green's function is used.
- The number of levels to specify is one.
- There is no conducting groundplane under the bottom layer.
- Layer one has infinite thickness, and
- The z-axis coordinate for the boundary between layers one and zero need to be set at some height higher than the structure of interest.

D.3 FEKO DI-card Setup

The method of using the DI-card to specify a dispersive medium is somewhat more circuitous than the GF-card method. The DI-card actually specifies the dielectric properties for a small dielectric cavity, with the normal vectors of the cavity surfaces pointing towards the volume of which the dielectric properties are assignable. The trick in using the DI-card to model a very large volume is that a small cavity is still specified, but with its normal vectors pointing towards the outside. This causes FEKO to model the entire area *around* the cavity as a dielectric. Once the small cavity has been defined, it is quite easy to specify the dielectric properties of the medium, using equation D.1.

D.4 Comparison between the GF-card and the DI-card

The question now arises as to which of these modelling options should be used. The decision to use the DI-card formulation was made on the following grounds:

- The GF-card method works very slowly compared to the DI-card method.
- Comparative results show that both methods approximate the reflection coefficient of a short dipole equally well. Figure D.1 shows a comparison of S_{11} between the GF-card method and the DI-card method.

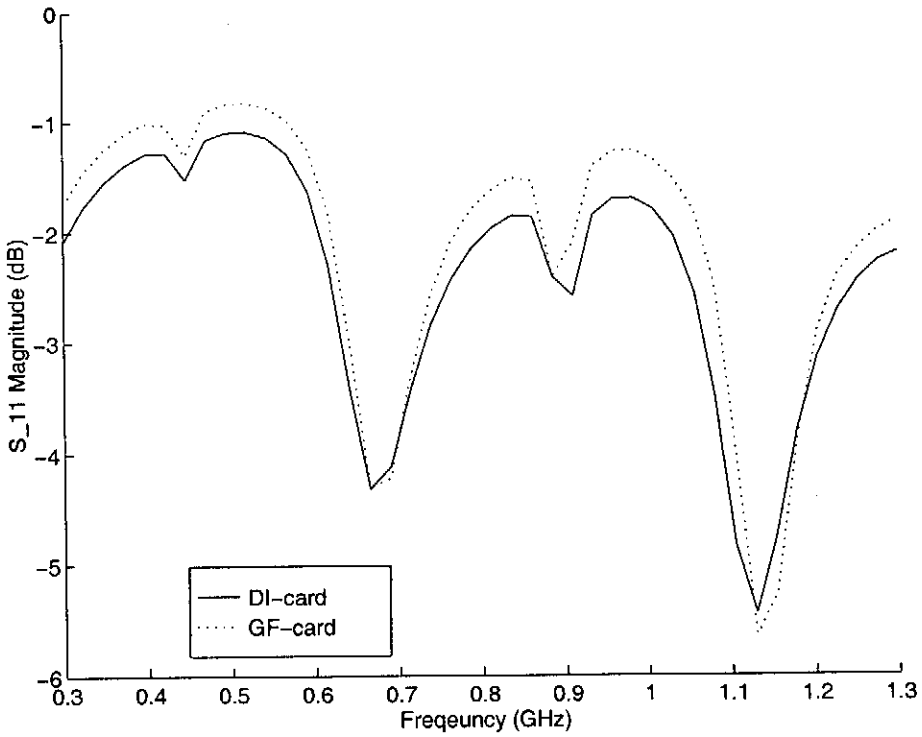


Figure D.1: S_{11} comparison for FEKO dispersive media models

Appendix E

OpenGL Visualization

One of the major drawbacks to using Fortran for scientific work is the lack of a native visualization ability in the Fortran language. Albert Einstein was quoted as saying: "If I can't picture it, I can't understand it." In other words, Fortran may be a good language to use for scientific work, but can not display the data in a graphical and easy to understand and interpret format.

This problem has been overcome of late with the National Institute of Standards and Technology's introduction of OpenGL bindings for Fortran-90, [37]. These bindings are easily installed onto most Fortran-90 development suites and are distributed with sufficient examples, carefully documented in [38], to enable quick visualization of data from inside Fortran[1] code. In this project OpenGL was used to create an ability for the FDTD code to instantly graph sampled field values. This ability speeds up development by reducing the amount of post-processing that needs to be done to visualize field values.

E.1 What is OpenGL?

Put quite simply: OpenGL is a software interface to graphics hardware. The interface consists of about 250 commands that are used to specify objects and operations that are required in your specific graphical implementation.

The key feature to working with OpenGL, is that the programmer only needs to know the geometry of the structure to visualize, and the rotations or translations that have to be performed on the image to display it in from the desired angle, [39]. With the proper specifications OpenGL handles the remaining matrix manipulations, projections and rendering necessary to place the image on the screen. As OpenGL was originally implemented with the goal of creating an interface that would work on a number of computing platforms, no generic shapes or modelling instructions are provided. Instead, OpenGL requires that all models are built up using a standard set of geometric primitives - points, lines and polygons.

[1] Refer to Appendix chapter F for a detailed description of how these bindings may be installed and used with Compaq Visual Fortran 6.1

OpenGL performs four basic functions to render an image on the screen:

- Constructs shapes from geometric primitives, thereby creating mathematical descriptions of objects.
- Arranges the objects in 3D space and selects the desired vantage point for viewing the structure.
- Calculates the colors of the specified objects. These calculations take into account the colors that were specified explicitly, but also the colour perturbations that are needed when using specific lighting or texture maps.
- Converts the mathematical description of the specified objects and their associated colors into pixels that are displayed on the screen. (Also referred to as *rasterization.*)

OpenGL is often referred to as a *state machine*, because of the many states that are set during its operation. Features that are controlled by states remain in that state until it is explicitly changed. Examples of states that are set are the current colour, current viewing and projection transformations, line and polygon stipple patterns and position and characteristics of light.

E.2 GLUT, the OpenGL Utility Toolkit

Since OpenGL was created to be independent of any specific windowing system or operating system, it does not contain functions for opening or managing windows or reading input from the keyboard or mouse. The GLUT solves this problem and also provides a number of useful functions to draw complicated 3D structures, such as spheres, toroids and even a teapot. It must be mentioned that GLUT probably falls a little short when full-featured OpenGL functions are to be created, but it is a good starting point for the creation of simple OpenGL functions to learn more about OpenGL.

Table E.1 presents a few of the GLUT windowing routines that were used in this thesis.

glutInit	Initializes the GLUT and should be called before any other GLUT routine
glutInitDisplayMode	Specifies whether to use a RGBA or colour-index colour model.
glutInitWindowPosition	Specifies the screen location for the upper left corner of the new window.
glutInitWindowSize	Specifies the size, in pixels, of the new window.
glutCreateWindow	Creates a window with an OpenGL context. The window is not displayed until **glutMainLoop** is called.

Table E.1: GLUT windowing functions

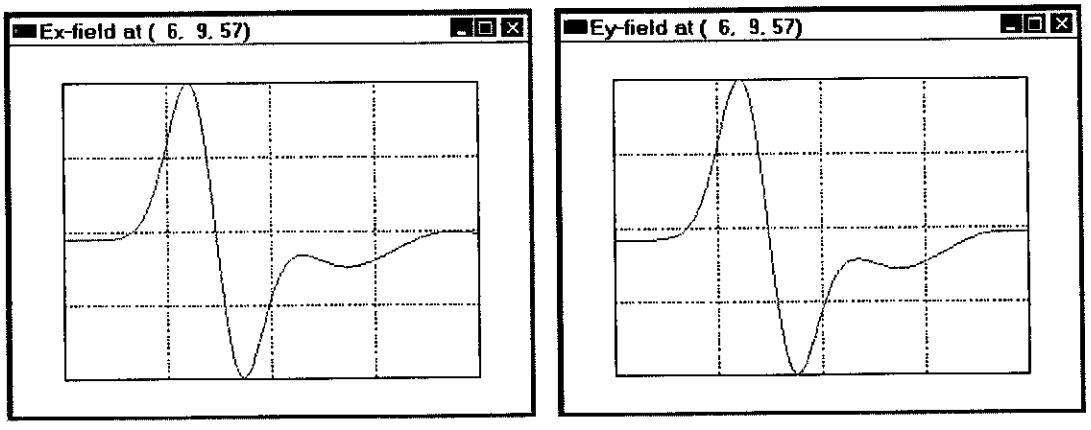
Apart from the windowing functions, the most important GLUT function that was used in this thesis is the display callback function, **glutDisplayFunc**. Whenever GLUT determines that a window needs to be redrawn, the callback function that was registered by **glutDisplayFunc** is executed. This callback function thus needs to contain all the instructions that are needed to redraw the specific window. Situations which require the redraw of a window include:

- When a window is brought to front from behind another window.
- When a window is dragged across the screen.
- When a window is resized.

Once all the necessary callback and windowing functions have been specified the final function that is required for the display of data is executed. **glutMainLoop** simply initiates event processing and triggers the registered display functions. It is important to know that **glutMainLoop** is actually an endless loop that is never exited. If a window is closed the entire loop is shut down and all other displays that are controlled with the GLUT are also shut down.

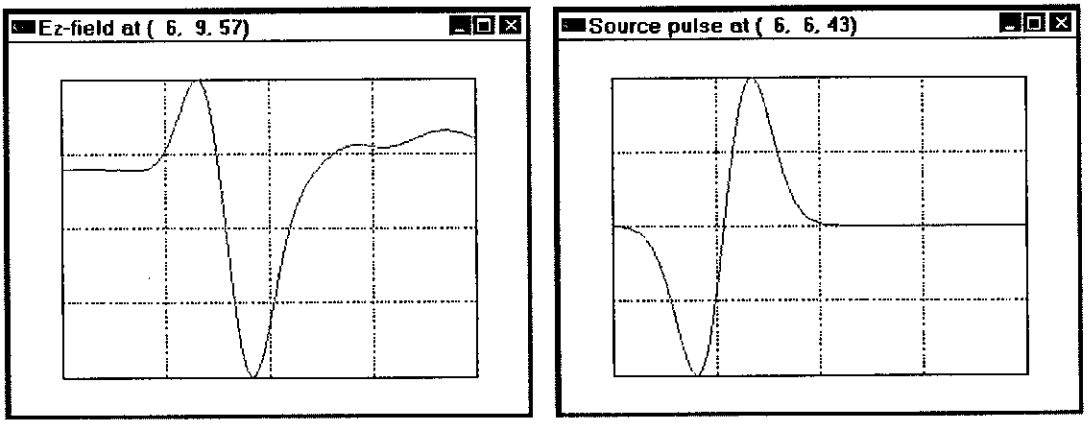
E.3 OpenGL Screenshots

Examples of what the OpenGL routines graph are given in figures E.1 and E.2.



(a) Ex (b) Ey

Figure E.1: OpenGL output windows - Examples 1,2



(a) Ez (b) Source

Figure E.2: OpenGL output windows - Examples 3,4

Appendix F

Compaq Visual Fortran 6.1 and OpenGL

F.1 Introduction to Fortran and OpenGL

Compaq Visual Fortran 6.1 is one of the good Fortran compilers around today. It lends itself to use in the field of Computational Physics and in this case Computational Electromagnetics. The main problem in using Fortran for projects in the aforementioned fields is the lack of visualization ability in native Fortran programming. This document explains how to obtain and install the OpenGL bindings for Compaq Visual Fortran and how to use them in visualization of Fortran data.

F.2 OpenGL standards and Software

The OpenGL standard and its interface to Fortran 90 is explained clearly in [37]. The object of this section is to explain how to install the required software to exploit the marriage between Compaq Visual Fortran and OpenGL.

The first step is to obtain the required files. The required files are contained in:

- F90GLI.zip, downloadable from
  `http://www.digital.com/fortran/dvf/f90gl.html`
- F90GLUT.zip, downloadable from
  `http://math.nist.gov/mcsd/Staff/WMitchell/f90gl/software.html`
- FGLEXAMP.zip, downloadable from
  `http://math.nist.gov/mcsd/Staff/WMitchell/f90gl/software.html`

Once these files are obtained they need to be unzipped and configured in the following manner.

F.2.1 F90GLI.zip

This file contains the required library (.LIB) and module (.MOD) files to implement the OpenGL standard for Compaq Visual Fortran. The library files should by copied to the Fortran directory `DF98\LIB` and the module files should be copied to the directory `DF98\INCLUDE`.

F.2.2 F90GLUT.zip

This file contains a precompiled Dynamically Linked Library (.DLL) file to help Windows implement the Graphics Library Utility Toolkit (GLUT) functions required by Fortran. It also contains a library (.LIB) file which defines the association between GLUT and Fortran. The DLL should be copied to the `\WINDOWS\SYSTEM` directory and the library file should be copied to `DF98\LIB` again.

F.2.3 FGLEXAMP.zip

This file can now be unzipped into a directory of its own. It contains several sample files which implements a variety of OpenGL features. It can be used to test the installation. It is customary to test the system by first running the "trivial.f90" example and then the others.

F.3 Use of OpenGL Extensions in Fortran

Although the installed files are all in directories that should be referenced directly by Windows and Fortran they should be added to the linker specification in the appropriate Fortran project as well. This is done as follows:

Go to *Project*, *Settings*, *Link*, *General* and add the following to the "Object/Library modules" window: *f90GL.lib f90GLUT.lib f90GLU.lib glut32.lib.*

These statements instruct Fortran to call the appropriate libraries when linking the project.

F.4 Final Notes

The installation procedure described above should work, but could possibly be outdated by changed internet addresses and filenames. In the event that the information seems to be outdated, please reference the National Institute of Standards website[1] and work from there. All information and files mentioned here originate from this site or links from this site.

[1] http://math.nist.gov/~mitchell

Appendix G

Code Fragments

G.1 Thin Wire Update Equations

```
SUBROUTINE HX_THINWIRE(i,j,k)
C Implement the Hx field updates for a thin wire with resistive or
C  capacitive loading.
INTEGER i,j,k
REAL Econst

C assume all spacial DELTAs are equal
Econst = 0.5*LOG(DELz/wire(i,j,k))
      HXS(i,j,k) = HXS(i,j,k)
     &          - DTMdy/Econst*(EZS(i,j+1,k)-EZS(i,j,k))
     &             + DTMdz*(EYS(i,j,k+1)-EYS(i,j,k))

END SUBROUTINE HX_THINWIRE
```

G.2 Impedance Loaded Dipole Update Equations

```
SUBROUTINE EZ_THINWIRE(i,j,k)
C Implement the Ez field updates for a thin wire with resistive or
C  capacitive loading.
INTEGER i,j,k
REAL Rconst, Cconst, Hconst

IF (RC_TYPE(i,j,k).NE.0) THEN
C     Dispersive media update equations
CALL EZ_THINWIRE_DISPERSIVE(i,j,k)

ELSE
C Freespace update equations
```

```
! Resistor position
     IF ( resist(i,j,k).NE.0.0 ) THEN
    Rconst = DT*DELz/EPS0/2.0/resist(i,j,k)/DELx/DELy
    Hconst = 1.0/(1.0 + Rconst)
    Rconst = (1.0 - Rconst)/(1.0 + Rconst)

    EZS(i,j,k) = Rconst*EZS(i,j,k)
     &       + Hconst*DTEdx*(HYS(i,j,k)-HYS(i-1,j,k))
     &    - Hconst*DTEdy*(HXS(i,j,k)-HXS(i,j-1,k))

! Capacitor position
  ELSEIF ( capac(i,j,k).NE.0.0 ) THEN
    Cconst = 1.0+(capac(i,j,k)*DELz/EPS0/DELx/DELy)
    Cconst = 1.0 / Cconst
    EZS(i,j,k) = EZS(i,j,k)
     &          + Cconst*DTEdx*(HYS(i,j,k)-HYS(i-1,j,k))
     &    - Cconst*DTEdy*(HXS(i,j,k)-HXS(i,j-1,k))
  ELSE
! Wire only - enforce Etan=0
       EZS(i,j,k) = 0.0
  ENDIF

ENDIF

END SUBROUTINE EZ_THINWIRE
```

G.3 Dispersive Media Update Equations

```
SUBROUTINE EX_DISPERSE(i,j,k)
C Update for Ex in disperive media
INTEGER i,j,k, matnum

! Define material type specified in cell
matnum = RC_TYPE(i,j,k)

! Update Recursive Convolution sum
RC_SUM(1,i,j,k) = EXS(i,j,k)*RC_SPEC(matnum,2)
     & + RC_SUM(1,i,j,k)*RC_SPEC(matnum,1)

! Update equation including all terms
EXS(i,j,k) = (RC_SPEC(matnum,4)*EXS(i,j,k) + RC_SUM(1,i,j,k)
     &  + (HZS(i,j,k)-HZS(i,j-1,k))*DTEdy
     &  - (HYS(i,j,k)-HYS(i,j,k-1))*DTEdz)
     &                          /RC_SPEC(matnum,3)
END SUBROUTINE EX_DISPERSE
```

Bibliography

[1] J. C. Trickett, I. Mason, H. Eybers, M. Meyering, F. Stevenson, D. Vogt, J. Hargreaves, and R. Fynn, "The application of borehole radar to South Africa's ultra-deep gold mining environment," in *8th Ground Penetrating Radar Conference 2000, Australia*, pp. 676–681, May 2000.

[2] J. K. Schweitzer and R. A. Johnson, "Geotechnical classification of deep and ultra-deep Witwatersrand mining areas, South Africa," *Mineralium Deposita*, vol. 32, pp. 335–348, 1997.

[3] K. L. Shlager and J. B. Schneider, "A selective survey of the finite-difference time-domain literature," *IEEE Transactions on Antennas and Propagation*, vol. 37, no. 4, pp. 39–56, 1995.

[4] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, pp. 302–307, March 1966.

[5] A. Taflove, *Computational Electrodynamics: The Finite Difference Time Domain Method.* Artech House, 1995.

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

[7] G. Mur, "Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations," *IEEE Transactions on Electromagnetic Compatibility*, vol. EMC-23, pp. 377–382, Nov. 1981.

[8] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *Journal of Computational Physics*, vol. 114, pp. 185–200, October 1994.

[9] B. Engquist and A. Majda, "Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic field equations," *Mathematics of Computation*, vol. 31, pp. 629–651, 1977.

[10] D. S. Katz, E. T. Thiele, and A. Taflove, "Validation and extension to three dimensions of the Berenger PML absorbing boundary condition for FD-TD meshes," *IEEE Microwave and Guided Wave Letters*, vol. 4, pp. 268–270, 1994.

[11] J.-P. Berenger, "A perfectly matched layer for freespace simulation in finite-difference computer code," *Annales des Telecommunications*, 1994.

[12] F. Akleman and L. Sevgi, "A novel implementation of Berenger's PML for FDTD applications," *IEEE Microwave and Guided Wave Letters*, vol. 8, pp. 324–326, October 1998.

[13] T. Uno, Y. He, and S. Adachi, "Perfectly matched layer absorbing boundary condition for dispersive medium," *IEEE Microwave and Guided Wave Letters*, vol. 7, pp. 264–266, September 1997.

[14] A. Taflove (editor), *Advances in Computational Electrodynamics: The Finite Time-Domain Method.* Artech House, 1998.

[15] W. C. Chew and W. H. Wheedon, "A 3D perfectly matched medium from modified Maxwell's equations with stretched coordinates," *IEEE Microwave and Guided Wave Letters*, vol. 7, pp. 599–604, 1994.

[16] C. M. Rappaport, "Perfectly matched absorbing boundary conditions based on anisotropic lossy mapping of space," *IEEE Microwave and Guided Wave Letters*, vol. 5, pp. 90–92, 1995.

[17] Z. S. Sacks, D. M. Kingsland, R. Lee, and J. F. Lee, "A perfectly matched anisotropic absorber for use as an absorbing boundary condition," *IEEE Transactions on Antennas and Propagation*, vol. 43, pp. 1460–1463, 1995.

[18] S. Keller, "Design of impedance loaded antennas for borehole radar applications," Master's thesis, University of Stellenbosch (South Africa) and University of Karlsruhe (Germany), June 2000.

[19] C. M. Butler, "The equivalent radius of a narrow conducting strip," *IEEE Transactions on Antennas and Propagation*, vol. AP-30, pp. 755–757, July 1982.

[20] S. Watanabe and M. Taki, "An improved FDTD model for the feeding gap of a thin-wire antenna," *IEEE Microwave and Guided Wave Letters*, vol. 8, pp. 152–154, Apr. 1998.

[21] M. Piket-May, A. Taflove, and J. Baron, "FDTD modeling of digital signal propagation in 3-D circuits with passive and active loads," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, pp. 1514–1523, August 1994.

[22] S. Keller, "On designing impedance loaded antennas." Personal Communique, June 2000.

[23] W. Sui, D. A. Christensen, and C. H. Durney, "Extending the two-dimentional FDTD method to hybrid electromagnetic systems with active and passive lumped elements," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, pp. 724–730, April 1992.

[24] S. Ramo, J. R. Whinnery, and T. van Duzer, *Fields and Waves in Communication Electronics.* John Wiley and Sons, 3rd ed., 1994.

[25] D. Slater, *Near-field Antenna Measurements.* Artech House, 1991.

[26] E. M. Software and Systems, "Feko user's manual: Suite 2.3," January 2000.

[27] A. C. Newell and M. L. Crawford, "Planar near-field measurements on high performance array antennas," Tech. Rep. NB SIR 74-380, National Bureau of Standards, Colorado, July 1974.

[28] M. Bingle, "FDTD modeling of dispersive dielectrics," Master's thesis, University of Stellenbosch, November 1995.

[29] O. P. Ghandi, B.-Q. Gao, and J.-Y. Chen, "A frequency-dependent finite-difference time-domain formulation for general dispersive media," *IEEE Transactions on Microwave Theory and Techniques*, vol. 41, pp. 658–665, April 1993.

[30] D. B. Davidson and R. W. Ziolkowski, "A connection machine (CM-2) implementation of a three-dimensional parallel finite difference time-domain code for electromagnetic field simulation," *International Journal of Numerical Modeling: Electronic Networks, Devices and Fields*, vol. 8, pp. 221–232, 1995.

[31] K. R. Umashankar, A. Taflove, and B. Beker, "Calculation and experimental validation of induced currents on coupled wires in an arbitrary shaped cavity," *IEEE Transactions on Antennas and Propagation*, vol. AP-35, pp. 1248–1257, November 1987.

[32] P. C. L. Yip, *High-Frequency Circuit Design and Measurement.* Chapman and Hall, 1990.

[33] D. A. Frickey, "Conversions between S, Z, Y, h, ABCD, and T parameters which are valid for complex source and load impedances," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, pp. 205–211, February 1994.

[34] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications.* Prentice Hall, third ed., 1996.

[35] D. G. Zill and M. R. Cullen, *Advanced Engineering Mathematics.* PWS Publishing Company, 1992.

[36] C. M. Furse and O. P. Gandhi, "Why the DFT is faster than the FFT for FDTD time-to-frequency domain conversions," *IEEE Microwave and Guided Wave Letters*, vol. 5, pp. 326–328, October 1995.

[37] W. F. Mitchell, "A Fortran 90 interface for OpenGL," Tech. Rep. NISTIR 6134, National Institute of Standards and Technology, Mathematical and Computational Sciences Division, Colorado, January 1998.

[38] M. Woo, J. Neider, T. Davis, and D. Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2.* Addison-Wesley, 1999.

[39] M. Carr, "Visualization with OpenGL: 3D made easy," *IEEE Transactions on Antennas and Propagation*, vol. 39, pp. 116–120, August 1997.

[40] M. Metcalf and J. Reid, *Fortran 90 Explained.* Oxford University Press, 1994.