Analysis of Electromagnetic Wave Propagation using the 3D Finite-Difference Time-Domain Method with Parallel Processing

WILLIAM J. BUCHANAN

A thesis submitted in partial fulfilment of the requirements of Napier University for the degree of Doctor of Philosophy

March 1996

Abstract

The 3D Finite-Difference Time-Domain (FDTD) method simulates structures in the time-domain using a direct form of Maxwell’s curl equations. This method has the advantage over other simulation methods in that it does not use empirical approximations. Unfortunately, it requires large amounts of memory and long simulation times. This thesis applies parallel processing to the method so that simulation times are greatly reduced. Parallel processing, though, has the disadvantage in that simulation programs require to be segmented so that each processor processes a separate part of the simulation. Another disadvantage of parallel processing is that each processor communicates with neighbouring processors to report their conditions. For large processor arrays this can result in a large overhead in simulation time.

Two main methods of parallel processing discussed: Transputer arrays and clustered workstations over a local area network (LAN). These have been chosen because of their relatively cheapness to use, and their widespread availability. The results presented apply to the simulation of a microstrip antenna and to propagation of electrical signals in a printed circuit board (PCB). Microstrip antennas are relatively difficult to simulate in the time-domain because they have resonant pulses. Methods that reduce this problem are discussed in the thesis.

The thesis contains a novel analysis of the parallel processing showing, using equations, tables and graphs, the optimum array size for a given inter-processor communication speed and for a given iteration time. This can be easily applied to any processing system.

Background material on the 3D FDTD method and microstrip antennas is also provided. From the work on the parallel processing of the 3D FDTD a novel technique for the simulation of the Finite-element (FE) method is also discussed.

Table of contents

CHAPTER 1

Introduction

1.1 Introduction

This thesis relates to work carried-out in the Department of Electrical, Electronic and Computer Engineering, at Napier University, between April 1990 and December 1995. The principle investigators were Dr. Naren Gupta, Professor John Arnold and myself. Dr. Naren Gupta and myself are members of staff within the Electrical, Electronic and Computer Engineering Department at Napier University and Professor John Arnold is a member of staff in the Department of Electronics and Electrical Engineering at Glasgow University.

This chapter states the objectives of the research and outlines the basic structure of the thesis. It also provides some background information on the great Scottish scientist James Clark Maxwell, whose equations form the basis for the rest of the report.

1.2 Objectives

The main objectives of the research were to:

- Investigate frequency- and time-domain methods in the simulation of electromagnetic propagation;
- Model the propagation of electrical signals within microstrip antennas and printed circuit boards (PCBs) using the three-dimensional (3D) Finite-Difference Time-Domain (3D FDTD) method;
- Determine the electrical characteristics of microstrip antennas and PCBs using the 3D FDTD method;
- Investigate the application of parallel processing to simulations using the 3D FDTD method.

1.3 Thesis Structure and Background

The thesis contains nine main chapters and four appendices. This chapter introduces the thesis and provides some background material.

Initial research work investigated commonly used electromagnetic field simulation methods. Chapter 2 discusses some of these methods, including the method of moments, the finite-element and the finite-difference method. This research showed that the 3D FDTD method was the most useful method in modern three-dimensional simulations and was well suited to parallel processing. The principle researchers in this research project were one of the first to propose the application of the 3D FDTD method to limited memory computers [1.1] and to parallel processing [1.2].

Chapter 3 discusses background theory on the 3D FDTD method as it applies to electromagnetic (EM) wave propagation within PCBs and also in microstrip antennas. This theory provides a basis for the critical appraisal of the 3D FDTD method and its application to the modelling of novel structures.

The 3D FDTD research work led to the development of a fully automated 3D FDTD package that runs on any type of computer system, whether it be a low-specification PC, a multi-processor system, or even a large super-computer. Appendix 1 discusses the usage of this package and the source code listing of the modeller is available over the Internet or directly from the author.

The basic theory of the package is based on work carried out by Yee [1.3] who was the first reseacher to propose the 3D FDTD and Tavlove, *et. al.* [1.4] who expanded these theories. The thesis applies these techniques to the simulation of electromagnetic wave propagation within and outside microstrips antenna and printed circuit boards. This work is based on Sheen, *et. al.* [1.5], Zang, *et. al.* [1.5] and Railton, *et. al.* [1.7] who applied the method to the simulation of microstrip circuit, Taflove, *et. al.* [1.8] who applied it to scattering problems, and Railton, *et. al.* [1.9] and Buchanan, *et al.* [1.10] who applied it to the simulation of electromagnetic radiation.

Chapter 4 contains background theory on microstrip antennas, which is one of the structures simulated in the research. It uses the theory documented in the James and Hall series of books on microstrip antennas [1.11]–[1.13]. This is used to support the analysis conducted in Chapter 7.

Chapter 5 contains novel work relating to the parallel processing of the 3D FDTD method with transputer arrays. Fusco, *et. al.* [1.14], [1.15], Excell, *et. al.* [1.16], [1.17] and Buchanan, *et. al.* [1.18]–[1.21] have all applied parallel processing to the FDTD method. Fusco used small transputer arrays to implement a parallel 2D FDTD problem based on diakoptics. Excell and Tinniswood have applied the method to the simulation of electromagnetic waves on human tissues. The chapter concludes by showing a novel processor synchronisation method which significantly reduces inter-processor communication and thus reduces simulation times. It also contains novel equations and graphs which contrast simulation times for differing transputer array sizes and differing inter-communication transfer speeds.

Chapter 6 discusses the parallel processing of the 3D FDTD method over clustered workstations connectied over by a local area network (LAN). Other researchers, such as Excell and Tinniswood [1.22] have also applied parallel processing to the 3D FDTD method using a Meiko CS-2 MPP (massively-parallel processors) computer and the KSR-1 ‘virtual shared computer’. These computers are specially designed for parallel processing and have high-speed data links between processors. The chapter derives novel equations and displays graphs which contrast simulation times for practical multi-workstations connected over a standard Ethernet network.

Chapter 7 discusses the simulation of radiation and propagation in a microstrip antenna and within a PCB. Balanis and Panayiotis [1.23] applied the 3D FDTD method to model and predict the radiation patterns of wire and aperture structures. Sheen, *et. al.* [1.24] showed how the 3D FDTD method applies to the simulation of a microstrip antenna and Buchanan, Gupta and Arnold [1.25]–[1.29] describe the application of the 3D FDTD method to the simulation of microstrip antennas.

Chapter 8 discusses the simulation of the propagation of electromagnetic pulses within and outside a printed circuit board (PCB). These simulations are important in the design of electronic systems as they must now comply with EC Electromagnetic Compatibility (EMC) regulations. Most current methods involve building prototypes of systems and testing them to determine if they meet the EMC regulations. A better solution is to simulate the system by computer and modify the design so that it complies with the regulations. It is the intention of this chapter to discuss the application of the 3D FDTD method to the simulation of EMC from a PCB. Railton, *et. al.* [1.30], Pothecary *et. al.* [1.31] and Buchanan, *et. al.* [1.32]–[1.34] have applied the 3D FDTD method to determine the radiation and cross-talk from PCBs.

Finally, Chapter 9 presents the main conclusions of the research. It is the intention of the author to show that the 3D FDTD method provides accurate results and that use parallel processing significantly reduces simulation times and / or increased modelling sizes.

1.4 James Clerk Maxwell

James Clerk Maxwell was born in Edinburgh in 1831 and rates amongst the greatest of all the scientists [1.35]. His importance to the physical sciences and engineering puts him on par with Isaac Newton, Albert Einstein, James Watt and Michael Faraday.

Michael Faraday and Joseph Henry, independently, were the first to discover electrical induction [1.36]. In 1855, Maxwell took Faraday's ideas and theories about dielectric media and lines of force and developed a mathematical relationship between them. This is known as Faraday's law and, in modern vector notation, is expressed in (1.1). Faraday's law shows that a changing magnetic field induces an electric field proportional to the rate of change.

Maxwell then further developed the ideas of Amphere and Gauss to produce two further equations, known as Amphere's law (1.2) and Gauss's law (1.3). Amphere's law shows that a current produces a magnetical field proportional to the total current and Gauss's law shows that the total electrical flux density from a closed surface equals the total change enclosed.

Maxwell then added another law (1.4) which shows that the magnetic flux density out of a closed surface is zero. These four equation express the basic laws of electricity and magnetism, and are commonly known as Maxwell's equations.

$$e\frac{d\boldsymbol{E}}{dt} + \boldsymbol{J} = \nabla \times \boldsymbol{H} \tag{1.1}$$

$$m\frac{d\boldsymbol{H}}{dt} = -\nabla \times \boldsymbol{E} \tag{1.2}$$

$$\nabla \cdot \boldsymbol{E} = \frac{r}{e} \tag{1.3}$$

$$\nabla \cdot \boldsymbol{H} = 0 \tag{1.4}$$

Before Maxwell's work, many scientists had observed the relationship between electricity and magnetism, but it was Maxwell, though, who finally derived the mathematical link between these forces. His four short equations described exactly the behaviour and interaction of electric and magnetic fields. From this work he also proved that all electromagnetic waves, in vacuum, travel at 300 000 km.s^{-1}. This, Maxwell recognised, was equal to the speed of light and from this he deduced that light was also an electromagnetic wave.

He then reasoned that the electromagnetic wave spectrum contained many invisible waves, each with its own wavelength and characteristic. Other scientists, such as Hertz and Marconi soon discovered these 'unseen' waves, including infra-red, ultra-violet and radio waves.

1.5 Electromagnetic Fields

Maxwell found that all electrical signals propagate with an electric field and an associated magnetic field. His equations showed that a change in the electric field with respect to time causes a change in the magnetic field with respect to distance. This change in magnetic field causes a change in the electric field, and so the wave propagates.

1.6 3D Finite-Difference Time-Domain (FDTD) Method

The FDTD method provides a direct solution to Maxwell's equations with little complexity. In formulation Maxwell's continuous equations convert into a discrete form [1.8]. A mathematical modeller or computer then solves this discrete form. It has the advantage over other methods in that it takes into account all fields (electric and magnetic) in a 3D model and is well suited to parallel processing.

Results from this type of simulation gives the electric and magnetic fields in steps of time. Frequency information can then be extracted using Fast Fourier Transform (FFT) techniques. From this the frequency response over a wide spectrum can be determined. Other modelling methods normally require different models and / or techniques for different frequency spectra.

1.7 References

[1.1] Buchanan WJ, Gupta NK, "Simulation of Three-Dimensional Finite-Difference Time-Domain Method on Limited Memory Systems", *International Conference on Computation in Electromagnetics*, IEE, Savoy Place, London, 1991.

[1.2] Buchanan WJ, Gupta NK, "Parallel Processing of the Three-Dimensional Finite-Difference Time-Domain Method", *National Radio Science Colloquium*, University of Bradford, 7-8 Jul. 1992.

[1.3] Yee K, "Numerical Solutions of Initial Boundary Value Problems involving Maxwell's Equations in Isotropic Media", *IEEE Ant. and Prop.*, vol. 33, May 1966, pp. 302-307.

[1.4] Taflove A and Brodwin M, “Numerical solution of steady state electromagnetic scattering problems using the time dependent Maxwell’s equations”, *IEEE MTT*, vol. 23, no. 1, Aug. 1975, pp. 623–630.

[1.5] Sheen D, Ali S, Abouzahra M, and Kong J, “Application of Three-Dimensional Finite-Difference Method to the Analysis of Planar Microstrip Circuits”, *IEEE MTT*, vol. 38, pp. 849–857, Jul. 1990.

[1.6] X Zang, J Fang and K Mei, “Calculations of the dispersive characteristics of microstrips by the FDTD method”, *IEEE MTT*, vol. 26, pp. 263–267, Feb. 1988.

[1.7] Railton C and McGeehan, “Analysis of microstrip discontinuities using the FDTD method”, *MWSYM 1989*, pp.1089–1012.

[1.8] A Taflove, “The Finite-Difference Time-Domain Method for Electromagnetic Scattering and Interaction Problems”, *IEEE Trans. Electromagnetic Compatibility*, vol. EMC–22, pp. 191–202, Aug. 1980.

[1.9] Railton CJ, Richardson KM, McGeehan JP and Elder KF, “The Prediction of Radiation Levels from Printed Circuit Boards by means of the FDTD Method”, *IEE International Conference on Computation in Electromagnetics*, Savoy Place, London, Nov. 1991.

[1.10] WJ Buchanan, NK Gupta, “Prediction of Electric Fields from Conductors on a PCB by 3D Finite-Difference Time-Domain Method”, *IEE’s Engineering, Science and Education Journal*, Aug. 1995.

[1.11] James JR & Hall PS, “Handbook of Microstrip Antennas”, *IEE Electromagnetic Waves Series*, No. 28, Peter Peregrinus, 1989. Vol. 1.

[1.12] James JR & Hall PS, “Handbook of Microstrip Antennas”, *IEE Electromagnetic Waves Series*, No. 28, Peter Peregrinus, 1989. Vol. 2.

[1.13] James JR, Hall PS and Wood C, “Microstrip antennas, theory and design”, *IEE Electromagnetic Waves Series*, No. 19, Peter Peregrinus, 1989.

[1.14] Merugu L and Fusco V, “Concurrent Network Diakoptics for Electromagnetic Field Problems”, *IEEE MTT*, vol. 41, no. 4, Apr. 1993, pp. 708–716.

[1.15] Fusco V, Merugu L and McDowall, “An Efficient Diakoptics-based Algorithm for Electromagnetic Field Mapping”, *IEE’s 1st International Conference in Electromagnetics*, Savoy Place, London, Apr. 1991.

[1.16] Excell PS and Tinniswood AD, “A FDTD Program for Parallel Computers”, *QMW 1995 Antenna Symposium*, Queen Mary and Westfield College, July 1995.

[1.17] Excell PS and Tinniswood AD, “Parallel Computation of Large-scale FDTD problems”, *IEE 3rd International Conference in Electromagnetics*, University of Bath, Apr. 1996.

[1.18] WJ Buchanan, NK Gupta “Parallel Processing of the Three-Dimensional Finite-Difference Time-Domain Method”, *National Radio Science Colloquium*, University of Bradford, 7-8 Jul. 1992.

[1.19] WJ Buchanan, NK Gupta, “Simulation of Electromagnetic Pulse Propagation in Three-Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques”, *Electrosoft ‘93*, Jul. 1993, Southampton.

[1.20] WJ Buchanan, NK Gupta, “Parallel Processing Techniques in EMP Propagation using 3D Finite-Difference Time-Domain (FDTD) Method”, *Journal of Advances in Engineering Software*, vol. 18, 3, 1993.

[1.21] WJ Buchanan, NK Gupta, “Prediction of Electric Fields in and around PCBs – 3D Finite-Difference Time-Domain Approach with Parallel Processing”, *Journal of Advances in Engineering Software*, Dec. 1995.

[1.22] PS Excell, AD Tinniswood, “A Finite-Difference Time-Domain Program for Parallel Computers”, *1995 Antenna Symposium*, Queen Mary & Westfield College, July 1995.

[1.23] Tirkas PA and Balanis CA, “Finite-Difference Time-Domain Method for Antenna Radiation”, *IEEE Trans. on Antennas and Propagation*, vol. 40, 3, pp 334–857, March 1992.

[1.24] Sheen D, Ali S, Abouzahra M, and Kong J, “Application of Three-Dimensional Finite-Difference Method to the Analysis of Planar Microstrip Circuits”, *IEEE MTT*, vol. 38, 7, pp. 849–857, July 1990.

[1.25] Buchanan WJ, Gupta NK, “Simulation of Near-Field Radiation for a Microstrip Antenna using the 3D-FDTD Method”, *NRSC ‘93*, University of Leeds, Apr. 1993.

[1.26] Buchanan WJ, Gupta NK and Arnold JM, “Simulation of Radiation from a Microstrip Antenna using Three-Dimensional Finite-Difference Time-Domain (FDTD) Method”, *IEE Eight International Conference on Antennas and Propagation*, Heriot-Watt University, Apr. 1993.

[1.27] Buchanan WJ, Gupta NK and Arnold JM, “3D FDTD Method in a Microstrip Antenna’s Near-Field Simulation”, *Second International Conference on Computation in Electromagnetics*, Apr. 1994.

[1.28] Buchanan WJ, Gupta NK and Arnold JM, “Application of 3D Finite-Difference Time-Domain (FDTD) Method to Predict Radiation from a PCB with High Speed Pulse

Propagation", *Ninth International Conference on Electromagnetic Compatibility*, University of Manchester, UK, Aug. 1994.

[1.29] Buchanan WJ, Gupta NK, "An Accurate Model for the Parallel Processing of the 3D Finite-Difference Time Domain (FDTD) Method in the Simulation of Antenna Radiation", *QMW 1996 Antenna Symposium*, Jul. 1995.

[1.30] Railton CJ, Richardson KM, McGeehan and Elder KF, "The Prediction of Radiation Levels from Printed Circuit Boards by means of the Finite-Difference Time-Domain Method", *International Conference on Computation in Electromagnetics*, Nov. 1991, pp. 339–341.

[1.31] Pothecary N and Railton CJ, "Rigorous analysis of cross-talk on high speed digital circuits using the Finite Difference Time Domain Method", *International Journal on Numerical Modelling*, part H, 6, pp. 368–374.

[1.32] Buchanan WJ, Gupta NK, "Simulation of Electromagnetic Pulse Propagation in Three-Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques", *Electrosoft '93*, Jul. 1993, Southampton.

[1.33] Buchanan WJ, Gupta NK and Arnold JM, "Application of 3D Finite-Difference Time-Domain (FDTD) Method to Predict Radiation from a PCB with High Speed Pulse Propagation", *Ninth International Conference on Electromagnetic Compatibility*, University of Manchester, Aug. 1994.

[1.34] Buchanan WJ, Gupta NK, "Prediction of Electric Fields from Conductors on a PCB by 3D Finite-Difference Time-Domain Method", *IEE Engineering, Science and Education Journal*, Aug. 1995.

[1.35] Hart M, "The 100: A ranking of the most influential persons in history", Simon and Schuster, 1993. (Note: James Clerk Maxwell is rated the 24th most influential person of all time, ahead of Karl Marx, Napleon Bonaparte and Ludwig van Beethoven).

[1.36] Atherton W, "From Compass to Computer: A History of Electrical and Electronic Engineering", San Francisco Press Inc., 1984.

CHAPTER 2

Simulation Methods

2.1 Introduction

Electromagnetic design and simulation involve representing a simulated system by a mathematical model. The type of model used normally depends on parameters such as the required accuracy, the total simulation time, the type of results required, the frequency bandwidth, and so on. For example, modelling a system for its DC and low frequency characteristics normally involves using electrical energy sources such as voltage and current sources, and components such as resistance, capacitance and inductance. Voltages and currents within the system are then determined using impedance calculations.

For very high frequency simulations the physical structure of the system, normally, affects its electrical characteristics. For example, a bend on a copper track causes a reduction in signal strength because some of the electromagnetic waves reflect back from the mismatch caused by the bend. At low frequencies this effect would be negligible because their relatively large wavelength. For example, in free-space a 50 Hz signal has a wavelength of 6 000 000 m; whereas at 10 GHz the wavelength is only 0.03 m. Large wavelengths are generally less restricted by physical objects and discontinuities, and are also less affected by other effects, such as the skin effect, electromagnetic coupling and so on.

The main methods used in high frequency electromagnetic wave simulations are ones that take into account changes in the physical and dielectric structure, these are:

- Finite-Difference Determination of Eigenvalues;
- Finite-Difference Time-Domain Method;
- Variational and Related Methods;
- Finite Element Method;
- Method of Moments;
- Spectral Analysis with Fourier Series and Fourier Integral;
- Transmission Line Matrix.

The principle application of these methods to electromagnetics is in guided waves, antenna modelling and scattering. Analysis of microstrip and similar transmission lines is more difficult because they have non-uniform dielectics and thus cannot support a TEM wave. This chapter discusses some of these methods.

2.2 Matrix Solutions

Many methods in electromagnetic field simulation involve the solution of equations as matrices. In many applications these matrices contain many zero terms. A sparse matrix is one that contains many zero terms, while a dense matrix contains mostly non-zero terms.

Techniques, such as Gauss-Seidel solve sparse matrix problems and pivoting methods solve dense matrices. Pivoting involves the interchange of rows or columns, while partial pivoting interchanges rows and full pivoting interchanges rows and columns.

2.3 Time Domain Versus Frequency Domain Simulations

The response of system defined by how it modifies an input signal. Mathematically, it is ratio of the output frequency signal divided by the input signal frequency.

In determining the response of a system, the electric or magnetic field within the model are monitored at input and output points. The location of these depend on the type of simulation conducted. For example, to determine the amount of reflected energy from a patch antenna, the input and output points are placed at the same location, that is, both would be placed at the source of the antenna. Whereas, if the radiation pattern from the patch antenna is to be determined, the input location would be placed at the feed of the antenna and the output at points around the antenna, as illustrated in Figure 2.1.

Normally, a system is simulated for its frequency response. The actual modelling of the system is usually simpler using frequency-dependent elements rather than with time-dependent elements. Thus, the system frequency response, `H(f)`, is determined by simply dividing the output frequency response, `O(f)`, by the input signal system response, `I(f)`, as illustrated in Figure 2.2.

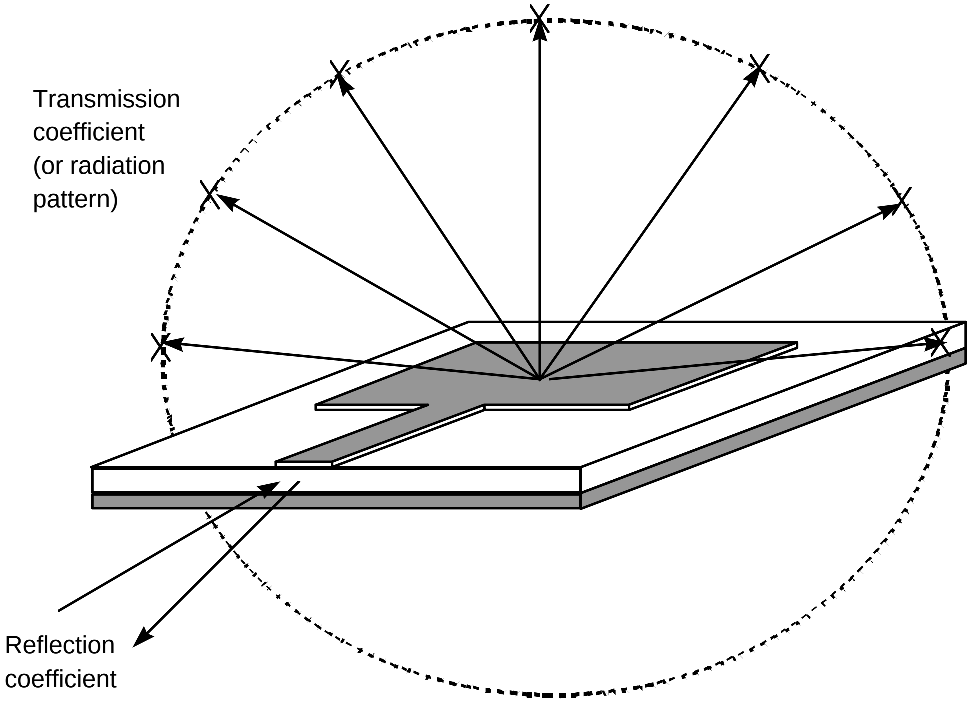


Figure 2.1: Monitoring of field within the model

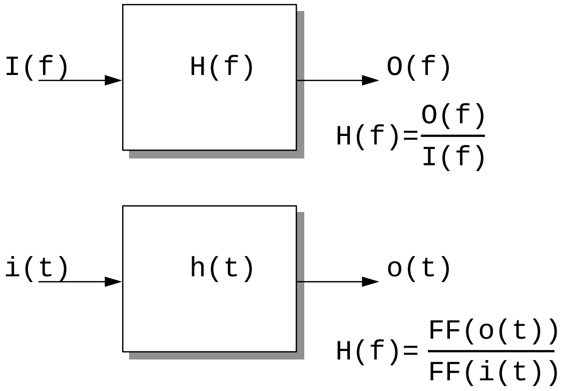


Figure 2.2: System response using frequency- and time-based signals

A discrete time-domain simulation involves stepping a system through increments of time. A fourier transform then converts the time-based input and output signal to give the frequency response for the input and output signal. The system frequency response is then the ratio of the output frequency response (`O(f)`) divided by the input frequency response (`I(f)`).

2.4 Converting from Continuous to Discrete

Electromagnetic field simulations normally involve the rate of change of electrical or magnetic fields with respect to distance or time, that is, first-order equations. They may also involve second-order equations that use the rate of change of the rate of change of the fields.

Some continuous equations can be solved if they have a standard form, but, unfortunately most real-life problems have no direct solutions. In these cases discrete equations can be made continuous form by approximations. The approximation can relate to time, frequency or physical dimensions.

For example, Figure 2.3 shows a continuous square function (*f(x)*=x^2), the partial different approximation is:

$$\frac{\P f}{\P x} = \frac{f(x+\Delta) - f(x-\Delta)}{2\Delta} \qquad (2.1)$$

thus, for *f(x)*=x^2:

$$\frac{\P f}{\P x} = \frac{(x+\Delta)^2 - (x-\Delta)^2}{2\Delta} \qquad (2.2)$$

for example if *x*=*2*, then

$$\left.\frac{\P f}{\P x}\right|_{x=2} = \frac{(2+0.1)^2 - (2-0.1)^2}{2 \times 0.1} = 4 \qquad (2.3)$$

Figure 2.4 shows how the second-order differential $\frac{\P^2 f}{\P x^2}$ is determined using the rate of changes, $\frac{\P f_1}{\P x}$ and $\frac{\P f_2}{\P x}$, thus

$$\frac{\P^2 f}{\P x^2} = \frac{\frac{f(x+\Delta)-f(x)}{\Delta} - \frac{f(x)-f(x-\Delta)}{\Delta}}{\Delta} \tag{2.4}$$

$$\frac{\P^2 f}{\P x^2} = \frac{f(x+\Delta)-2f(x)+f(x-\Delta)}{\Delta^2} \tag{2.5}$$

In general, the smaller the value of *D*, the more accurate the calculation of the differential will be.

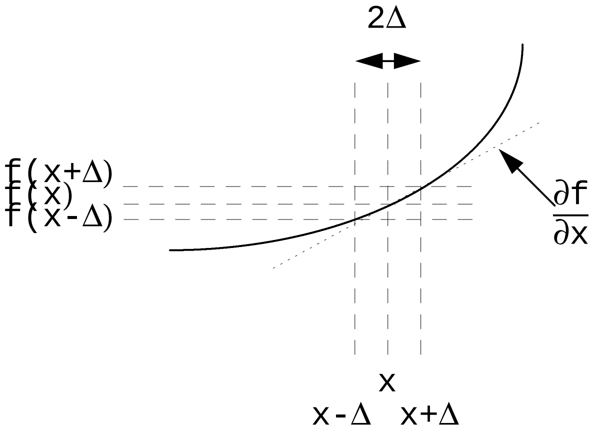


Figure 2.3: Determining first-order function

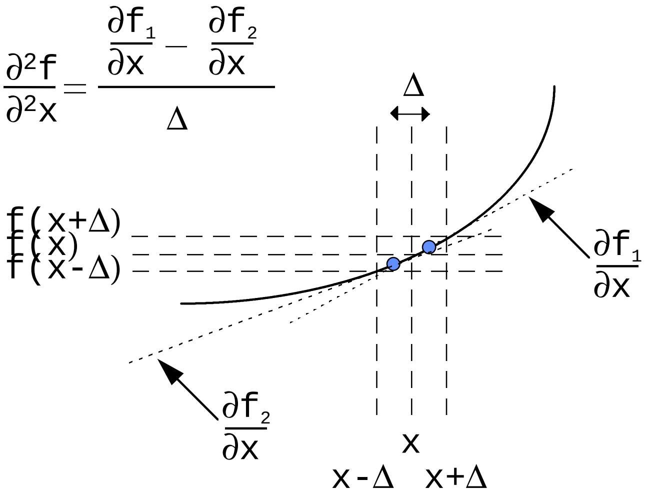

$\frac{\partial^2 f}{\partial^2 x} = \frac{\frac{\partial f_1}{\partial x} - \frac{\partial f_2}{\partial x}}{\Delta}$
Δ
f(x+Δ)
f(x)
f(x-Δ)
$\frac{\partial f_1}{\partial x}$
$\frac{\partial f_2}{\partial x}$
x
x-Δ
x+Δ

Figure 2.4: Determining second-order function

2.5 Two-Dimensional Modelling versus Three-Dimensional

Three-dimensional (3D) modelling takes into account changes in the physical structure in all three dimensions, whereas, two-dimensional modelling makes the approximation that the structure is unchanging in the dimension that the slice is taken through. Figure 2.5 shows an example of a 2D model. In this case a slice is taken through the *y*-*z* plane and thus does not take into account any changes of structure in the *x*-direction. For this reason 3D modelling is normally used where there is a non-uniform physical or electrical structure (non-homogenous).

Unfortunately, 3D models lead to greater simulation times. For example, if a structure splits into sub-elements and if each element takes the same time to simulate, then, a simulation with a 100×100×100 grid takes 100 times longer than for an equivalent 2D model. The simulation time is likely to be even greater than this estimate as, normally, 2D model elements provide a faster solution than 3D elements because they only require calculations for each of the sides of the element while 3D elements may require calculations for each of the faces of the element and possibly for each of the three vectors in space.

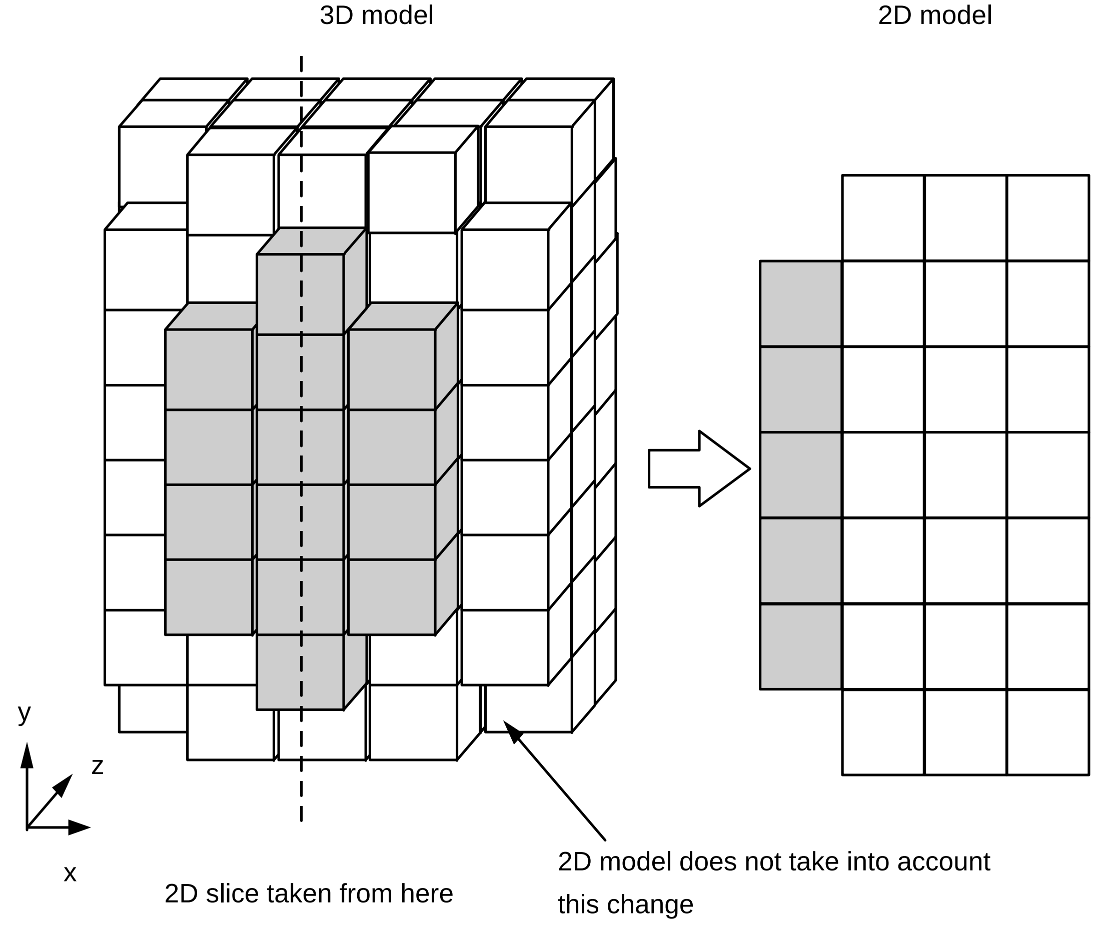


Figure 2.5: 3D and 2D modelling

2.6 Simulation Methods

This section discusses the main simulation methods and has been included in order to understand the advantages and disadvantages of the simulation methods used in the thesis. Figure 2.6 shows that the main methods split into four main areas, these are:

- Volume elements methods;
- Surface element methods;
- Ray methods;
- Hybrid methods.

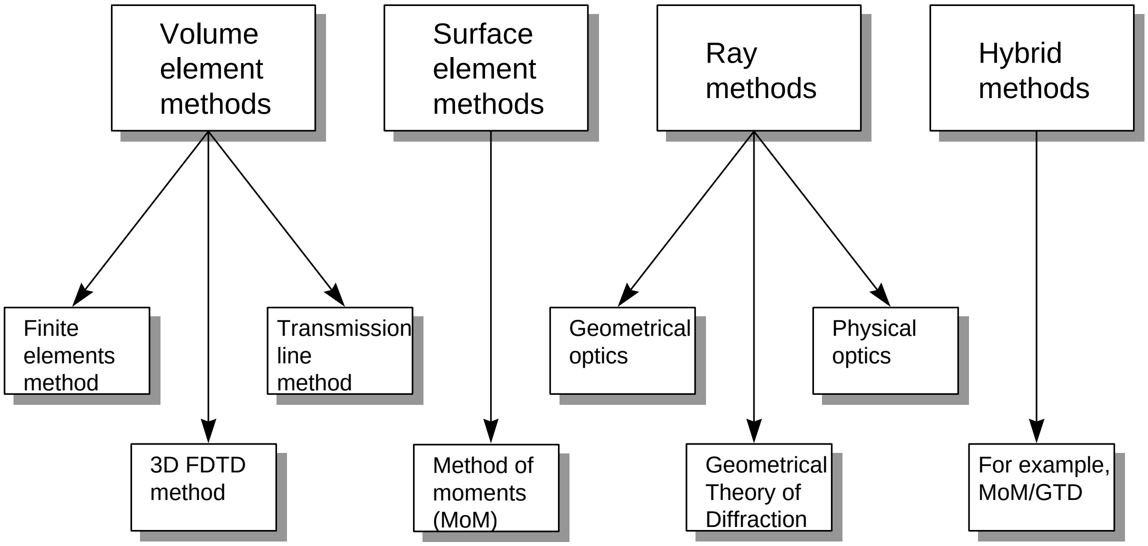


Figure 2.6: Modelling methods

2.6.1 Volume element methods

Volume element methods rely upon 3D subdivision, or elements, of the system together with material descriptions. Figure 2.7 shows an example of a structure converted into a number of elements. The modelling of each element may vary from element to element.

Finite-element (FE) method

The finite-element method splits the physical structure into smaller elements which are made from relatively simple shapes, such as cubes (for 3D) and squares or triangles (for 2D). These elements are modelled with an electromagnetic wave propagating through it, or, in some cases, modelled by discrete electrical components. The complete model is then built by connecting the inputs and outputs of the elements to their neighbours.

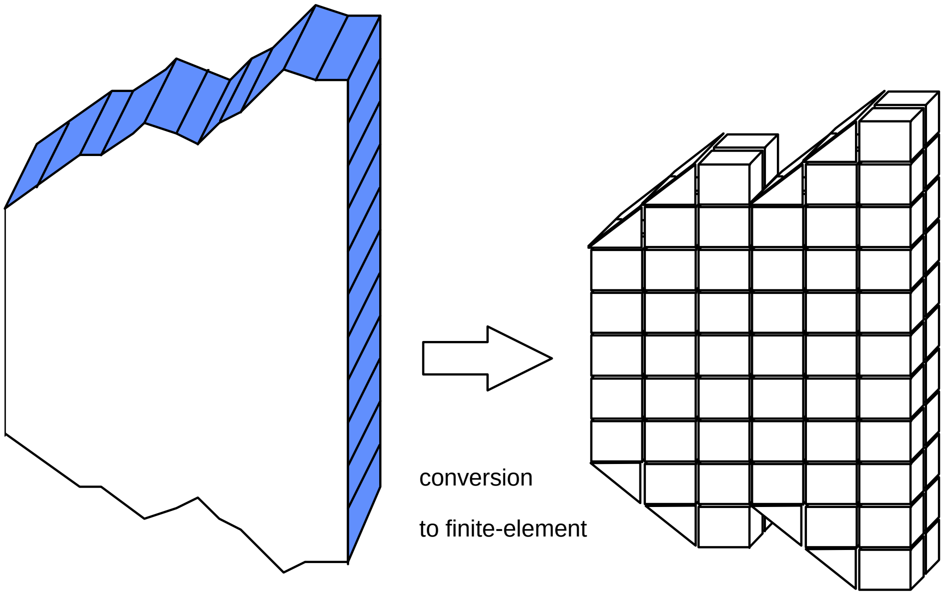


Figure 2.7: Conversion to finite-element

Finite-difference time-domain (FDTD)

Maxwell's equations define the relationship between the electric field and the magnetic field and are:

$$m\frac{d\boldsymbol{H}}{dt} = -\nabla \times \boldsymbol{E} \tag{2.6}$$

$$e\frac{d\boldsymbol{E}}{dt}+\boldsymbol{J}=\nabla\times\boldsymbol{H} \tag{2.7}$$

$$\nabla\cdot\boldsymbol{E}=\frac{r}{e} \tag{2.8}$$

$$\nabla\cdot\boldsymbol{H}=0 \tag{2.9}$$

Equations (2.6) and (2.7) show that a change in the electric field (***E***) produces a change in magnetic field (***H***), and vice-versa. Equation (2.8) shows that the electric field (***E***) relates to the electric charge (r) and Equation (2.9) shows that the magnetic field relates to magnetic charge (although no magnetic charge exists).

The 3D FDTD method is a time-domain simulation and involves stepping the system though discrete periods of time to give a transient response. Frequency information is then extracted using a fast fourier transform.

The main advantage of the FDTD method is that it provides a direct solution to Maxwell’s curl equations without much complexities. It also takes into account electric and magnetic fields in a three-dimensional model which other empirical analytical methods do not.

Chapter 3 discusses the 3D FDTD method in more detail and chapters 6 and 7 show the application of parallel processing to the method. Chapters 8 and 9 then apply it to model electromagnetic propagation within and outside microstrip antennas and PCBs. Unfortuately, microstrip antennas are highly resonant structures and thus, as the FDTD method is time-based it requires long simulation times. Chapters 6 and 7 discuss techniques which reduce this problem, and also methods to improve accuracy. These methods include sub-gridding around discontinuities and the application of parallel processing.

Transmission line matrix (TLM)

The TLM method is a time-domain method where an electromagnetic wave propagates through elements made from transmission lines. As with the FDTD method, a fast fourier transform convents the transient response into frequency response data [2.1]. It is variation of the finite-difference method but the boundary splits into elements rather than the interior region. The element used consist of a network of interconnected transmission lines [2.2].

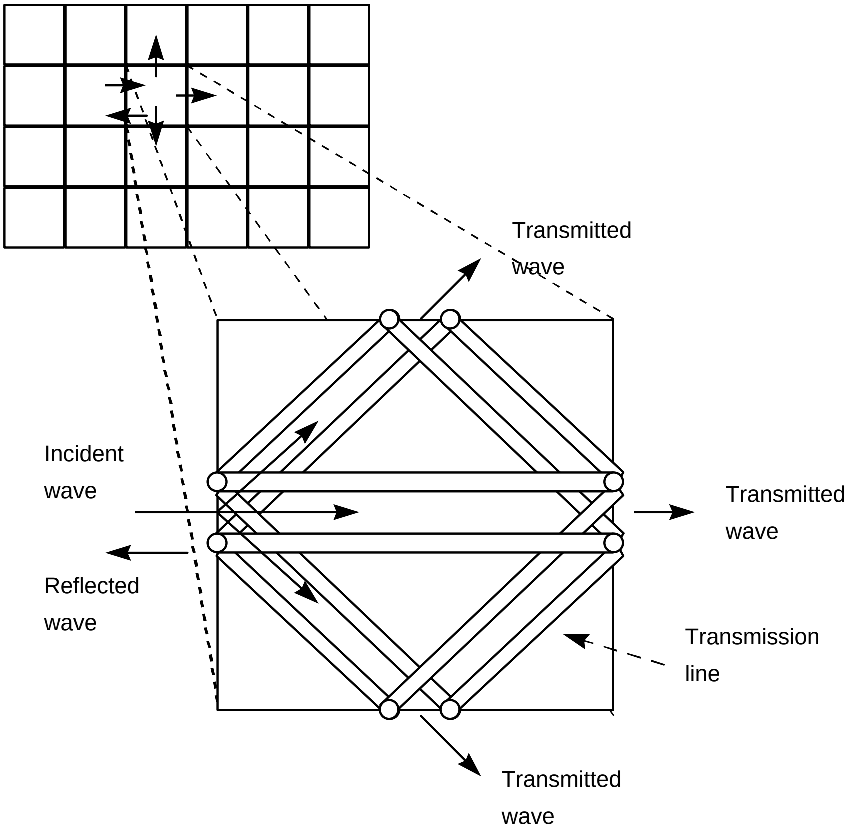


Figure 2.8: 2D TLM modelling

Figure 2.8 shows an example of a 2D element with 4 ports. The applied wave travels through the structure and is scattered by each of the lines within the element. These scattered waves then travel into neighbouring elements.

The TLM method accounts for material properties and boundaries by setting the properties of the transmission line. Systems with transverse electric (TE) and transverse magnetic (TM) modes have two equivalent transmission lines for each mode. Thus, in 3D model, there are 12 ports on each element.

The method has advantages similar to the 3D FDTD method, in that, it takes into account both the electric and magnetic fields in a 3D model, and, because it is a time-based simulation, it produces a wide-bandwidth response. It is also relatively straight-forward to implement and different physical structures can be modelled using non-linear grids – these include hybrid variable meshes [2.3], multi-grid meshes [2.4] and general curvilinear co-ordinates [2.5].

Refer to Christopoluos [2.6] and Hoefer [2.7] for more information on the TLM method.

Mode matching (MM)

Mode matching divides a system into a number of inter-connected sections. If each of the sub-sections has a solution involving known modes, then the complete system can be analysed by enforcing continuity of tangential field components at the interfaces between the sub-sections.

Figure 2.9 shows an example of change of width of a rectangular waveguide carrying a TE_{01} mode. The **E** and **H** fields on each side of the interface can easily expand to give an infinite series of modal functions. Equating the tangential field components at the interface and terminating the summations to a finite number of terms yield approximate equalities.

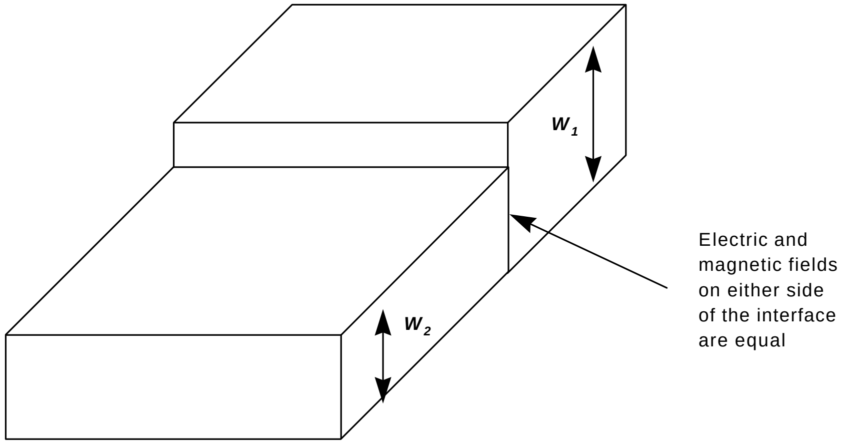


Figure 2.9: Mode matching within a waveguide

2.6.2 Surface elements methods

In surface element methods, the electric and magnetic fields do not penetrate into the elements [2.9], whereas, volume methods compute the fields within the element. Surface element methods generally require much less elements than volume elements methods, but material properties are difficult to define.

The method of moments (MoM) is a surface element method and is one of the most widely used computational methods in electromagmetics. When dielectrics are used with the MoM it turns the problem into a volume element problem. This leads to an increase in complexity and runtime. Fortunately, special techniques can be used in certain cases to alter the properties of the surface elements to take into account material changes. Unfortunately, these techniques increase the simulation time. An alternative formulation for a surface element is to use a wire grid with equivalent radii for the surface.

Method of moments

Finite-difference methods, typically, solve differential equations, whereas moment methods solve integral equations. For example, Poisson's equation, relates a scalar potential (*V(x,y,z)*) to the electric charge (*r(x,y,z)*), expressed mathematically as:

$$\nabla^2 V(x,y,z) = -\frac{r(x,y,z)}{e} \tag{2.10}$$

As an integral equation the scalar potential at a separation distance *R* becomes:

$$V(x,y,z) = \int_v \frac{r(x,y,z)}{4pe_0|R|}\, dv \tag{2.11}$$

In the moments method the unknown function is under the integral sign. In this case, the general form is:

$$V(x,y,z) = \int_v K(x,y,z) r(x,y,z) dv \tag{2.12}$$

where *K(x,y,z)* is the kernel of the equation. *V(x,y,z)* and *K(x y,z)* are known, but the function *r(x,y,z)* is unknown. The method of moments then determines the unknown variable.

First the total charge distribution is found by summing the individual charge contributions of *N* incremental sub-volumes forming the region under consideration, thus:

$$r(x,y,z) = \sum_{i=1}^{N} K_i f_i(r) \tag{2.13}$$

where K_i are, as yet, unknown constants and $f_i(r)$ are, as yet, unknown functions. For example if the voltage is a constant within a confined space then:

$$V = \int \frac{r_v dv}{4per} \tag{2.14}$$

It can be shown from [2.8] that this leads to a matrix equation in the form:

$$[B] = [A] \cdot [r] \tag{2.15}$$

where

$$[B] = \begin{bmatrix} V_1 \\ V_2 \\ . \\ . \\ V_N \end{bmatrix}, [A] = \begin{bmatrix} A_{11} & . & . & . & A_{1N} \\ . & & & & . \\ . & & & & . \\ . & & & & . \\ A_{N1} & . & . & . & A_{NN} \end{bmatrix} \text{ and } [r] = \begin{bmatrix} r_{11} \\ r_{21} \\ . \\ . \\ r_{N1} \end{bmatrix} \tag{2.16}$$

Cramer’s rule, matrix inversion or Gaussian elimination then determines the array [ρ]. The solution can be found by solving for [ρ] to give:

$$[r] = [A]^{-1}[B] \tag{2.17}$$

2.6.3 Ray methods

Ray methods involve tracing the path of rays when they are reflected or diffracted from an object. The amount and direction of diffraction and reflection depends upon the type of surface (both geometrical and material). There are solutions for a wide range of conducting materials but only for a limited amount for dielectric problems.

Ray tracing dominates the simulation time and is often difficult to estimate. The computation of geodesics on general curved surfaces can be very time consuming. Fortunately, unlike the finite-element and finite-difference methods, data storage is not normally a problem. The main methods used are:

- geometrical theory of diffraction (GTD)
- physical optics (PO);
- geometrical optics (GO).

These methods are general and only applied, in electromagnetics for the simulation of reflective antennas. The GO method is satisfactory for aperture diameters which are large in terms of wavelength. As the reflector aperture decreases, the radiation patterns become increasingly dominated by edge diffraction.

2.6.4 Hybrid methods

Hybrid methods involve a mixture of two or more of the volume, surface or ray tracing methods.

2.7 Conclusions

Mode matching is generally useful when modelling simple waveguide structures, but, cannot be applied to complex structures or resonant simulations.

The FE, FDTD and TLM methods split structures into groups of interconnected elements. The FE method is frequency-based and models each of the elements with their equivalent frequency characteristics.

The FDTD and TLM methods are time-based and involve stepping a model through discrete intervals in time. They differ in the way that they model the elements. The TLM method models uses equivalent transmission line for each element, whereas, the FDTD method models the propagation through the elements using a discrete form of Maxwell's curl equations.

As has been stated, the 3D FDTD method provides a direct solution to Maxwell's curl equations and takes into account both the electric and magnetic fields in a three-dimensional model. Other analytical methods, such as the TLM method, use empirical approximations.

Time-domain simulations have the disadvantages over frequency-based in that they normally require relatively long simulations times and that structures may not be easily modelled as time-based models. Modern computers overcome the first problem because they have large amounts of memory storage and have fast processor speeds. The FDTD method overcomes the second problem because it derives directly from Maxwell's equations.

In general, time-domain solutions have the advantages over frequency-domain solution in that that they provide wide bandwidth responses and they can be used in parallel processing with reduced simulation times (this will be discussed in more detail in Chapters 6 and 7). For these reasons the 3D FDTD method has been chosen as the main simulation method in this research.

2.8 References

[2.1] Krumpholz M, Russer P, “On the Dispersion in TLM and FDTD”, *IEEE Transactions on MTT*, vol. 42, no. 7, pp. 1277–1279, July 1994.

[2.2] Simpson N and Bridges E, “Equivalence of propagation characteristics for the TLM and FDTD”, *IEEE Transactions on MTT*, vol. 39, pp. 354–357, Feb. 1991.

[2.3] Scaramuzza R, Lowery AJ, *Electronic Letters*, no. 26, pp. 1947–1948

[2.4] Herring J.L, Christopolous C, *Electronic Letters*, no. 27, pp. 1794–1795.

[2.5] Meliani H, de Cogan D, Johns PB, *International Journal of Numerical Modelling*, 1988, pp. 221–238.

[2.6] Christopolous C, Field Analysis Software based on the transmission-line modelling method, *Advances in Engineering Software*, Springer-Verlag, pp. 135–148.

[2.7] W Hoefer, “The transmission line matrix (TLM) method”, ed. T. Itoh, *Numerical techniques in microwave and millimetre wave passive devices*, Wiley, 1981, pp. 496–591.

[2.8] Fusco V, “Microwave Circuits: Analysis and Computer-aided Design”, Prentice-Hall, 1987, pp. 130–142.

[2.9] Harrington, ”Field Computations by Moment Methods”, MacMillan, 1968.

CHAPTER 3

The 3D-FDTD Method

3.1 Introduction

The processing power and memory capacity of modern computers increases by the year. This has made possible the simulation of electromagnetic field problems in the time-domain rather than in the frequency-domain. Another change in simulation techniques has been from continuous equations to discrete approximations. These discrete forms are usually easier to implement on a computer.

A good example of a time-domain simulation, using discrete equations, is the 3D FDTD method. It determines the frequency response over a wide spectrum of frequencies, whereas many other simulation methods require different models and/or techniques for different frequency spectra. Papers [3.1]–[3.12] outline the basic theory and application of the 3D FDTD method.

The 3D FDTD method derives directly from Maxwell's curl equations and is relatively simple to implement. Unfortunately, it requires large amounts of computer memory and processing time and, has, in the past, only been used with super-computers which have the processing power and memory capacity to apply it. With the arrival of high-speed desktop computers with large and cheap memory storage the method can now be fully exploited in the areas such as microstrip antenna modelling, analysis of microstrip circuits and in biological applications.

The 3D-FDTD method has two main advantages over empirical analysis. It provides a direct solution to Maxwell's equations without much complexity and takes into account both the electric and magnetic fields in a 3D model.

As the 3D FDTD method is time-based the results produced can also help to provide an understanding of EM wave propagation within the structures. Frequency-domain techniques often conceal how the EM waves propagate within the structure. For example, a microstrip antenna (or patch antenna) can be modelled as a transmission line, or as lumped parameters. This modelling can often hide the fact that the incident waves within the antennas head are reflected back and forward within the antenna. Electromagnetic radiation then leaks out from the ends of the patch. Results from 3D FDTD simulations allow the wave to be visualised, which helps in checking results.

This chapter discusses the theory of the 3D FDTD method and its application to electromagnetic wave propagation within microstrip antennas and PCBs. Chapters 8 and 9 apply the 3D FDTD method and show some 3D pulse visualisations.

3.2 Background

Yee [3.1] was the first researcher to propose a modified form of the TLM method, which is now known as the FDTD method. It has since been used to model microstrip circuits [3.2]–[3.7], to scattering problems [3.8] and in the simulation of electromagnetic radiation [3.9], [3.10]. Other researchers have applied it to the simulation of waveguides, to coaxial cable and simular structures [3.11]–[3.16], and to digital signal processing and ferrites [3.17]–[3.20]. It is also useful in areas, such as in Biomedical research to model electromagnetic radiation on human tissues and to radar wave simulations.

3.3 Simulation Steps

Figure 3.1 shows the main steps taken in a 3D FDTD simulation. Initially, a 3D model is made to represent the physical structure, including conductors, dielectics and boundaries. Next an applied pulse, normally either a sine-wave or a Gaussian pulse, acts as the input stimulus at all the sources. Then at increments of time the **E** and **H** fields are calculated. After each increment the input electric field amplitude is calculated and the **E** and **H** fields are again recalculated. This continues until the **E** and **H** fields within the system decay to zero.

After completing the simulation an FFT program extracts frequency information from the transient response. The location of the transient data depends on the required system response. For example to determine the reflection coefficient, the input and reflected waves at the sources are monitored. For a radiation pattern, points are taken in free-space around the structure.

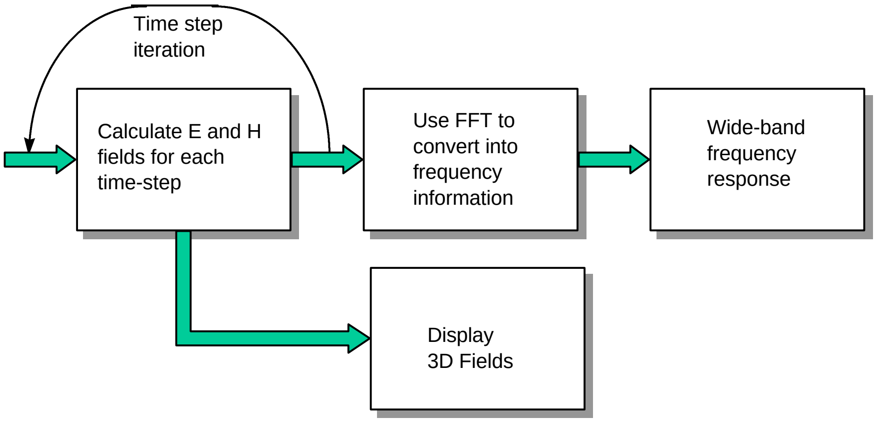


Figure 3.1: FDTD method

3.4 Finite-Difference Time-Domain (FDTD) Method

The FDTD method uses Maxwell’s equations which define the propagation of an electromagnetic wave and the relationship between the electric and magnetic fields, these are:

$$m\frac{d\boldsymbol{H}}{dt} = -\nabla \times \boldsymbol{E} \tag{3.1}$$

$$e\frac{d\boldsymbol{E}}{dt} + \boldsymbol{J} = \nabla \times \boldsymbol{H} \tag{3.2}$$

$$\nabla \cdot \boldsymbol{E} = \frac{r}{e} \tag{3.3}$$

$$\nabla \cdot \boldsymbol{H} = 0 \tag{3.4}$$

For a uniform, isotropic and homogeneous media with no conduction current Maxwell’s curl equations then become:

$$m\frac{d\boldsymbol{H}}{dt} = -\nabla \times \boldsymbol{E} \tag{3.5}$$

$$e\frac{d\boldsymbol{E}}{dt} = \nabla \times \boldsymbol{H} \tag{3.6}$$

By applying appropriate boundary conditions on sources, conductors and mesh walls an approximate solution of these equations can be found over a finite three-dimensional domain. Taking an example of the first equation in the ***i*** direction gives:

$$m\frac{\Delta H_x}{\Delta t} = \frac{\Delta E_y}{\Delta z} - \frac{\Delta E_z}{\Delta y} \tag{3.7}$$

The central difference approximation is then used on both the time and space first-order partial differentiations to obtain discrete approximations. This gives:

$$m\frac{H^{n+1/2}_{xi,j,k} - H^{n-1/2}_{xi,j,k}}{\Delta T} = \frac{E^{n}_{yi,j,k} - E^{n}_{yi,j,k-1}}{\Delta z} - \frac{E^{n}_{zi,j,k} - E^{n}_{zi,j-1,k}}{\Delta y} \tag{3.8}$$

Rearranging gives:

$$H^{n+1/2}_{x\,i,j,k} = H^{n-1/2}_{x\,i,j,k} + \frac{\Delta t}{m\Delta z}\left[E^{n}_{y\,i,j,k} - E^{n}_{y\,i,j,k-1}\right] - \frac{\Delta t}{m\Delta y}\left[E^{n}_{z\,i,j,k} - E^{n}_{z\,i-1,j,k}\right] \tag{3.9}$$

The half time-steps indicate that ***E*** and ***H*** are calculated alternately to obtain central differences for the time derivatives. In total there are six equations similar to Equation (3.9). These define the ***E*** and ***H*** fields in the *x*, *y* and *z* directions and are given in Equations (3.10) and (3.11). The permitivity (*e*) and the permeability (*m*) values in these equations are set to approximate values depending on the location of each of the field component.

$$\begin{aligned}
H^{n+1/2}_{x\,i,j,k} &= H^{n-1/2}_{x\,i,j,k} + \frac{\Delta t}{m\Delta z}\left[E^{n}_{y\,i,j,k} - E^{n}_{y\,i,j,k-1}\right] - \frac{\Delta t}{m\Delta y}\left[E^{n}_{z\,i,j,k} - E^{n}_{z\,i-1,j,k}\right] \\
H^{n+1/2}_{y\,i,j,k} &= H^{n-1/2}_{y\,i,j,k} + \frac{\Delta t}{m\Delta x}\left[E^{n}_{z\,i,j,k} - E^{n}_{z\,i,j,k-1}\right] - \frac{\Delta t}{m\Delta z}\left[E^{n}_{x\,i,j,k} - E^{n}_{x\,i-1,j,k}\right] \\
H^{n+1/2}_{z\,i,j,k} &= H^{n-1/2}_{z\,i,j,k} + \frac{\Delta t}{m\Delta y}\left[E^{n}_{x\,i,j,k} - E^{n}_{x\,i,j,k-1}\right] - \frac{\Delta t}{m\Delta x}\left[E^{n}_{y\,i,j,k} - E^{n}_{y\,i-1,j,k}\right]
\end{aligned} \tag{3.10}$$

$$\begin{aligned}
E^{n+1}_{x\,i,j,k} &= E^{n}_{x\,i,j,k} + \frac{\Delta t}{e\Delta y}\left[H^{n+1/2}_{z\,i,j+1,k} - H^{n+1/2}_{z\,i,j,k}\right] - \frac{\Delta t}{e\Delta z}\left[H^{n+1/2}_{y\,i,j,k} - H^{n+1/2}_{y\,i-1,j,k}\right] \\
E^{n+1}_{y\,i,j,k} &= E^{n}_{y\,i,j,k} + \frac{\Delta t}{e\Delta z}\left[H^{n+1/2}_{x\,i,j,k+1} - H^{n+1/2}_{x\,i,j,k}\right] - \frac{\Delta t}{e\Delta z}\left[H^{n+1/2}_{z\,i,j,k+1} - H^{n+1/2}_{z\,i,j,k}\right] \\
E^{n+1}_{z\,i,j,k} &= E^{n}_{z\,i,j,k} + \frac{\Delta t}{e\Delta x}\left[H^{n+1/2}_{y\,i+1,j,k} - H^{n+1/2}_{y\,i,j,k}\right] - \frac{\Delta t}{e\Delta y}\left[H^{n+1/2}_{x\,i,j+1,k} - H^{n+1/2}_{x\,i,j,k}\right]
\end{aligned} \tag{3.11}$$

3.5 Problem Conception

The structure simulated in Chapter 9 is a PCB with four electrical sources, as shown in Figure 3.2. It consists of a substrate layer, such as Duroid (relative permittivity, ε_r, of 2.2) above a ground plane. A copper layer is formed by etching the top of the substrate to give the required pattern.

3.5.1 3D gridding

A 3D grid is placed around the structure, as illustrated in Figure 3.3. The number of cells within the grid is normally selected with consideration to the simulation time limit and the amount of computer memory. An example linear grid placed around a microstrip antenna contained within a volume of 30×30×9.6 mm^3 with a 100×100×12 grid gives a element volume of 0.3×0.3×0.8 mm^3.

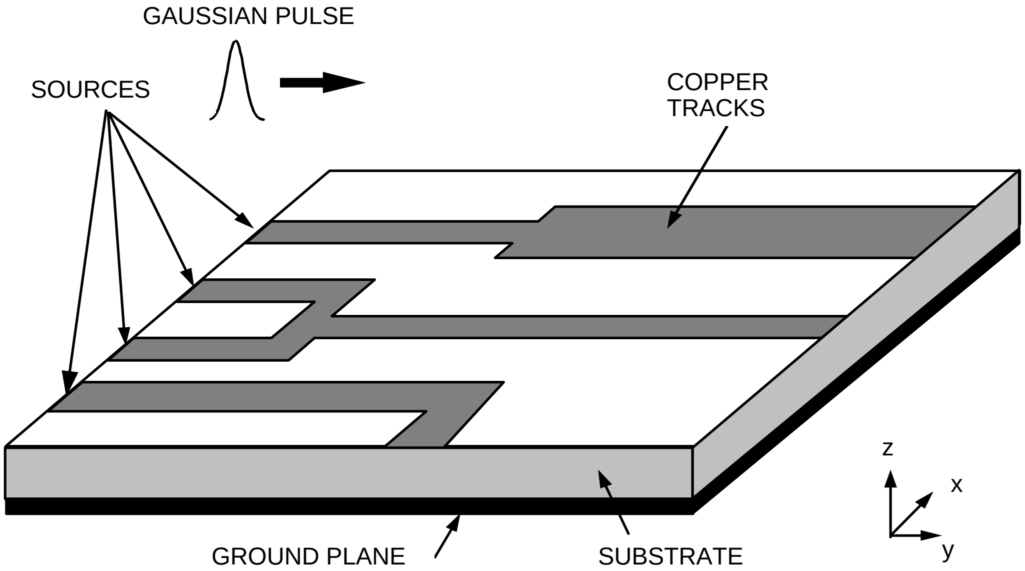


Figure 3.2: PCB with copper tracks

The first grid point in the *z*-direction lies on the top of the ground plane. Normally, there are fewer cells in the *z*-direction because there are very few discontinuities in this direction. A discontinuity causes reflections in the electromagnetic wave and they have a great effect on the frequency characteristics of the simulated model. Thus, to provide higher accuracy around discontinuities, a non-linear grid is sometimes placed around them. A fine grid is placed around discontinuities and a course grid where there are no discontinuities.

3.5.2 Permittivity and permeability

The calculation of the magnetic fields involves permeability. As conductors are assumed to have zero thickness, the value of μ_r is always taken as 1 (thus the permeability μ will be $4\pi\times10^{-7}$ H m^{-1}).

The calculation of electric fields uses permittivity which varies depending on whether the field is within the substrate or in the surrounding air. The permittivity in the medium above the substrate is $e_{r1}e_0$, and within the substrate it is $e_{r2}e_0$, (where e_0 is 8.854×10^{-12} F.m^{-1}). At the interface between the air and the substrate, the approximate relative permittivity is taken to be the average of the two, that is:

$$\frac{e_{r1}+e_{r2}}{2} \tag{3.12}$$

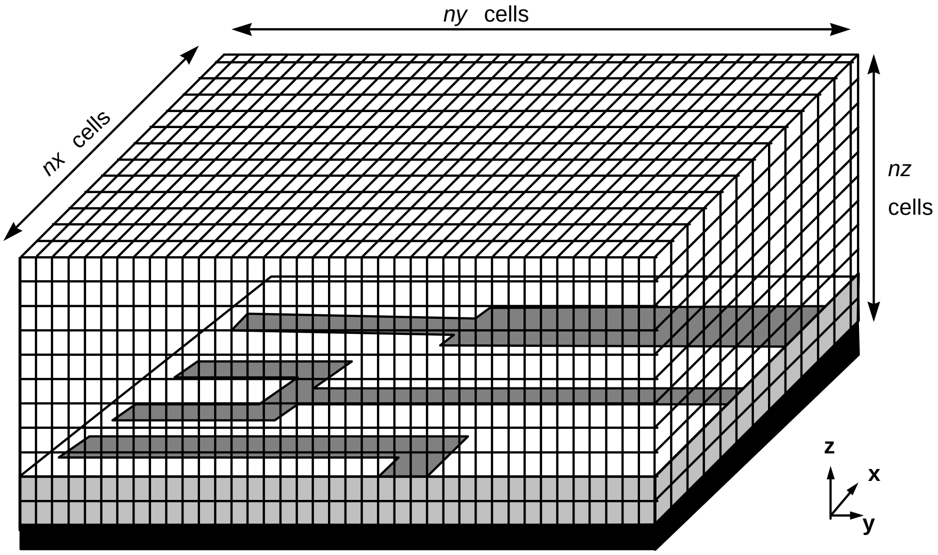


Figure 3.3: 3D gridding

3.5.3 Input signal

The input signal can be of any shape, but, it is normally a Gaussian pulse. This type of pulse has a frequency spectrum that is also Gaussian and thus has the advantage of providing frequency information from DC up to a desired cut-off frequency. The form of the input signal in a continuous form is:

$$f(t) = e^{-\frac{(t-t_0)^2}{T}} \tag{3.13}$$

where t_0 is the pulse delay and T relates to the pulse width. Written in a discrete form gives:

$$f(nT_S) = e^{-\frac{(nT_S - mT_S)^2}{xT_S}} = e^{-\frac{(n-m)^2}{x}} \tag{3.14}$$

where n is the current time-step, m the pulse delay time-step and x the width of the pulse in time-steps. Figure 3.4 shows Gaussian pulses with pulse widths of 5, 10 and 20 time-steps. Each pulse has been delayed by 30 time steps.

The width of the Gaussian pulse sets the required cut-off frequency. Figure 3.5 shows the relative power of a Gaussian pulse width of 5, 10 and 15 time steps. In can be seen that the thinner the pulse the larger its signal bandwidth.

The pulse width is normally chosen to have at least 20 points per wavelength at the highest frequency significantly represented in the pulse. In most cases in this thesis the pulse width is 11 time-steps, which gives a bandwidth of 20 GHz.

Initially in the simulation, all the electric and magnetic fields are set to zero. The Gaussian pulse applied at the source has only a field component which is perpendicular to the ground plane (that is, E_z). Thus, E_y and E_x, at the source, are always zero. A change in the electric field at the source with respect to time causes a change in the magnetic field in the *x*-direction. Thus, the wave propagates in the *y*-direction, as shown in Figure 3.6.

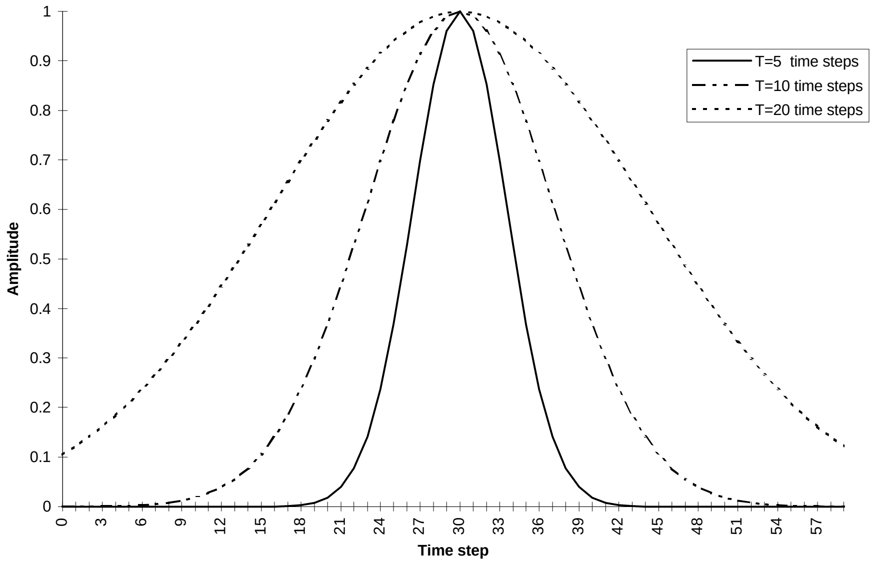


Figure 3.4: Gaussian pulse

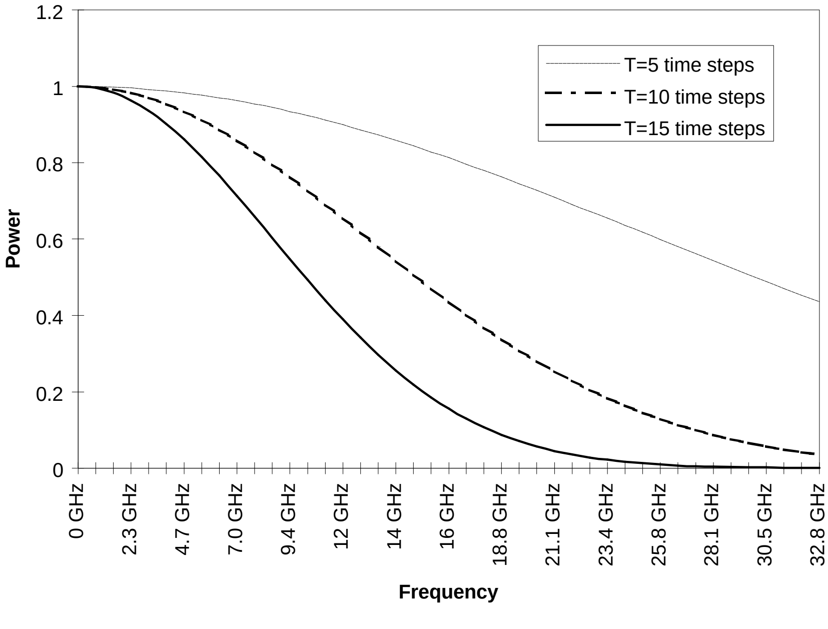


Figure 3.5: Gaussian pulse

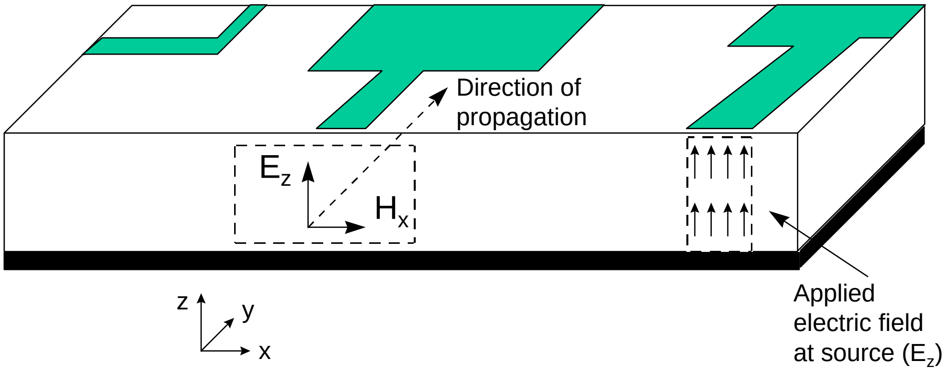


Figure 3.6: Propagation of the wave

3.5.4 Conductors

The 3D FDTD method assumes perfect electrical conductors. Thus, the tangential electric field components that lie on the conductors are assumed to be zero. Figure 3.7 shows that the E field components on the conductor will be zero in the *x*- and *y*- direction.

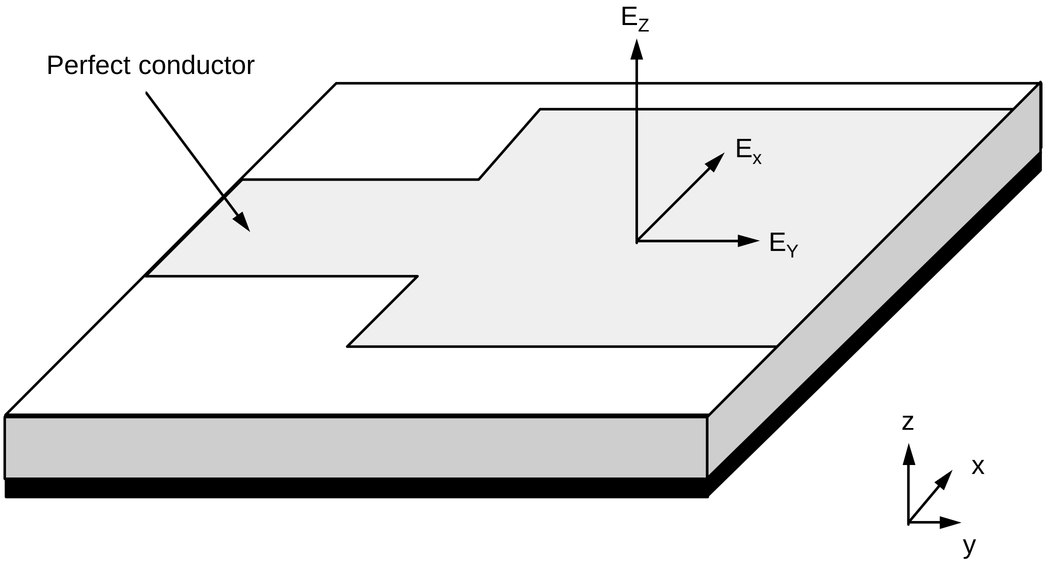


Figure 3.7: Conductor treatment

3.5.5 Boundary walls

There is a limit to the size of physical grid applied around the model. To reduce the requirements for a large grid an absorbing wall is placed on the six mesh boundary walls. The ground plane and its tangential electric fields are always zero and the tangential electric fields on the other five mesh walls are calculated so that a wave propagating against them does not reflect back. For the structure simulated in this thesis the pulses are normally incident on the mesh walls. This leads to simple approximations for continuous absorbing boundary conditions. The tangential fields on the absorbing boundaries then obey the one-dimensional wave equation in the direction normal to the mesh wall. For the normal y-direction wall the one-dimensional wave equation may be written as:

$$\left(\frac{\partial}{\partial y}-\frac{1}{v}\frac{\partial}{\partial t}\right)E_{\tan}=0 \qquad (3.15)$$

This equation is Mur's [3.21] first approximate absorbing boundary condition and in a discrete form it is:

$$E_0^{n+1}=E_1^{n+1}+\frac{v\Delta t-\Delta y}{v\Delta t+\Delta y}\left(E_1^{n+1}-E_0^{n}\right) \qquad (3.16)$$

where E_0 represents the tangential electric field on the mesh wall and E_1 the electric field one grid point within the mesh wall. Similar equations can also be derived for the other four absorbing mesh walls.

The method, unfortunately, does not take into account fringing fields that are propagating tangential to the walls. Thus the absorbing boundary must be placed well away from any fringing fields.

3.5.6 Maximum time step

The maximum time step that may be used is limited by the stability restriction of the finite difference equations [3.2]. This is given by:

$$\Delta t \leq \frac{1}{c}\left[\frac{1}{\Delta x^{2}}+\frac{1}{\Delta y^{2}}+\frac{1}{\Delta z^{2}}\right]^{-\frac{1}{2}} \tag{3.17}$$

where c is the speed of light (300 000 000 m.s^{-1}) and *Dx*, *Dy* and *Dz* are the dimensions of the unit element. Table 3.1 gives two example time steps for different element sizes.

Table 3.1: Example time intervals

Model size (mm^3)	*nx*, *ny*, *nz* elements	*Dx* (mm)	*Dy* (mm)	*Dz* (mm)	*Dt* (picoseconds)
30×10×10	100, 100, 12	0.3	0.3	0.3	0.68
80×80×50	100, 100, 10	0.8	0.8	0.5	1.25

3.6 Extracting Frequency Data

A fourier transform extracts frequency information from the transient response. Figure 3.8 shows an example of the electric field at a source. It can be seen that initially the Gaussian pulse is applied at the input. Then, after the pulse reaches the head of the antenna, a pulse reflects back to the source from the interface between the antenna head. Reflected pulses also return back from the resonance with the antenna head. The reflection coefficient is then the ratio of the reflected wave divided by the applied wave. In general, the scattering parameters S_{jk} may be obtained using a fourier transform on the transient waveforms, thus:

$$S_{jk}(w) = \frac{FF(V_j(t))}{FF(V_k(t))} \tag{3.18}$$

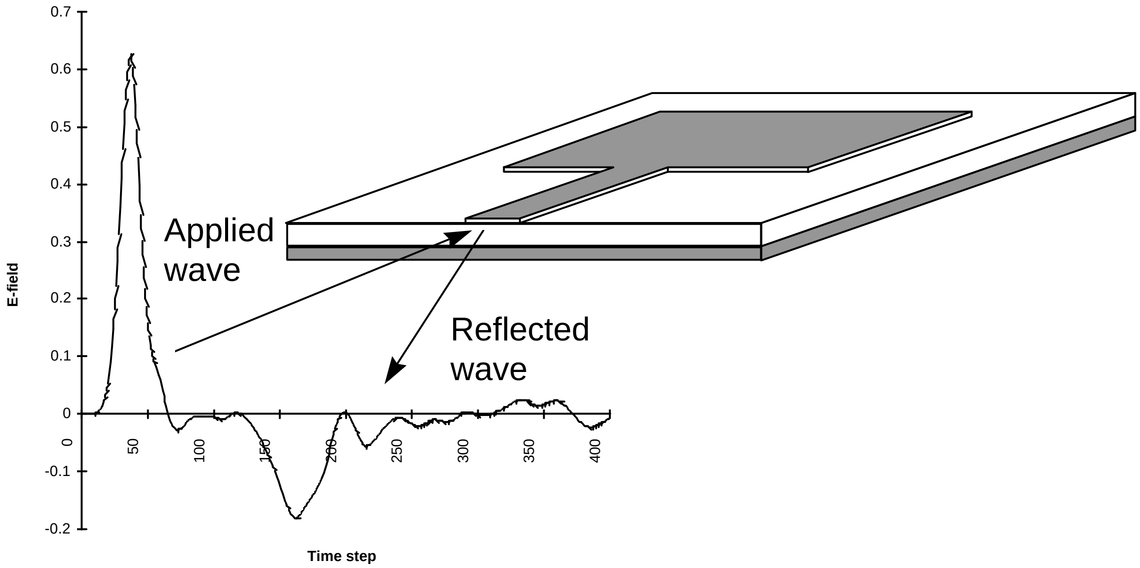


Figure 3.8: Applied and reflected wave

3.7 Improvements to the FDTD Method

Improvements can be made to FDTD which can improve accuracy, such as the sub-gridding method around discontinuities [3.22] and a modified frequency domain Finite-Difference Method that condenses nodes and uses an image principle [3.23]. Other researchers have incorporation of static field solutions and Z-transforms into the FDTD method [3.24]–[3.26].

Improvements can also be made to the boundary conditions and the modelling of sources [3.27]–[3.29]. Simulations times can be reduced using parallel processing methods [3.30],[3.31]. These will be discussed in more detail in chapters 6 and 7.

3.8 References

[3.1] Yee K, "Numerical Solutions of Initial Boundary Value Problems involving Maxwell's Equations in Isotropic Media", *IEEE Ant. and Prop.*, vol. 33, May 1966, pp. 302-307.

[3.2] Taflove A and Brodwin M, "Numerical solution of steady state electromagnetic scattering problems using the time dependent Maxwell's equations", *IEEE MTT*, vol. 23, no. 1, Aug. 1975, pp. 623–630.

[3.3] D Sheen, S Ali, M Abouzahra, and J Kong, "Application of Three-Dimensional Finite-Difference Method to the Analysis of Planar Microstrip Circuits", *IEEE MTT*, vol. 38, pp. 849–857, Jul. 1990.

[3.4] X Zang, J Fang and K Mei, "Calculations of the dispersive characteristics of microstrips by the FDTD method", *IEEE MTT*, vol. 26, pp. 263–267, Feb. 1988.

[3.5] Railton C and McGeehan, "Analysis of microstrip discontinuities using the FDTD method", *MWSYM 1989*, pp.1089–1012.

[3.6] Shibata T, Havashi T and Kimura T, "Analysis of microstrip circuits using three-dimensional full-wave electromagnetic field analysis in the time-domain", *IEEE MTT*, vol. 36, pp. 1064–1070, Jun. 1988.

[3.7] Feix N, Lalande M and Jecko B, "Harmonically Characterization of a Microstrip Bend via the FDTD Method", *IEE Proceedings*, ", *IEEE MTT*, vol. 40, no. 5, May 1992, pp. 955–961.

[3.8] A Taflove, "The Finite-Difference Time-Domain Method for Electromagnetic Scattering and Interaction Problems", *IEEE Trans. Electromagnetic Compatibility*, vol. EMC–22, pp. 191–202, Aug. 1980.

[3.9] Railton CJ, Richardson KM, McGeehan JP and Elder KF, "The Prediction of Radiation Levels from Printed Circuit Boards by means of the FDTD Method", *IEE International Conference on Computation in Electromagnetics*, Savoy Place, London, Nov. 1991.

[3.10] WJ Buchanan, NK Gupta, "Prediction of Electric Fields from Conductors on a PCB by 3D

Finite-Difference Time-Domain Method", *IEE's Engineering, Science and Education Journal*, Aug. 1995.

[3.11] Hese J and Zutter D, "Modelling of Discontinuities in General Coaxial Waveguide Structures by the FDTD-Method", *IEEE MTT*, vol. 40, Mar. 1992.

[3.12] Paul D, Pothercary and Railton, "Calculation of the Dispersive Characteristics of Open Dielectric Structures by the FDTD Method", *IEEE MTT*, vol. 42, no. 7, Jul. 1994.

[3.13] Navarro E, Such V, Gimeno B and Cruz J, "T-Junction in Square Coaxial Waveguide: A FDTD Approach" , *IEEE MTT*, vol. 42, no. 2, Feb. 1994, pp. 347–350.

[3.14] Moglie F, Rozzi T and Marcozzi P, "Wideband Matching of Waveguide Discontinuities by FDTD Methods", *IEEE MTT*, vol. 42, no. 11, Nov. 1994, pp. 2093–2098.

[3.15] Navarro A and Nuñez M and Martin E, "FDTD FFT method applied to axially symmetrical electromagnetic resonant devices", *IEE Proceedings*, vol. 137, pt. H, no. 3, Jun. 1990, pp. 193–196.

[3.16] Navarro A and Nuñez M, "FDTD Method Coupling with FFT: A Generalization to Open Cylinder Devices", *IEEE MTT*, vol. 42, no. 5, May 1994, pp. 870–874.

[3.17] Picket-May Melinda, Taflove A and Baron J, "FDTD Modelling of Digital Signal Processing in 3D Circuits with Passive and Active Loads", *IEEE MTT*, vol. 42, no. 8, Aug. 1994, pp. 1514–1523.

[3.18] Paul D, Pothercary and Railton, "Calculation of the Dispersive Characteristics of Open Dielectric Structures by the FDTD Method", *IEEE MTT*, vol. 42, no. 7, Jul. 1994.

[3.19] Wu K, Wu C and Litva J, "An Application of FDTD Method for Studying the Effects of Packages on the Performance of Microwave and High Speed Digital Circuits", *IEEE MTT*, vol. 42, no. 10, Oct. 1994, pp. 2007–2009.

[3.20] Pereda J, et al, "FDTD Analysis of Magnetized Ferrites: Application of the Calculation of Dispersion Characteristics of Ferrite-Loaded Waveguides", *IEEE MTT*, vol. 43, no. 2, Feb. 1995, pp. 350–356.

[3.21] Mur G, "Absorbing Boundary Conditions for the FDTD Approximation of the Time Domain Electromagnetic Field Equations", *IEEE EMC*, vol. 23, no. 2, Feb. 1981, pp. 377–382.

[3.22] Svetlana V, Yee K and Mei K, "A Subgridding Method for the Time-Domain Finite-Difference Method to Solve Maxwell's Equations", *IEEE MTT*, vol. 39, no. 3, Mar. 1991.

[3.23] Afande M, Giroux M and Bosisio R, "A FD Frequency Domain Method that Introduces

Condensed Nodes and Image Principle", *IEEE MTT*, vol. 43, no. 4, Apr. 1995.

[3.24] Shorthouse D and Railton C, "The Incorporation of Static Field Solutions Into the FDTD Algorithm", *IEEE MTT*, vol. 40, no. 5, May 1992, pp. 986–994.

[3.25] Prescott D and Shuley, "Reducing Solution Time in Monochromatic FDTD Waveguide Simulations", *IEEE MTT,* vol. 42, no. 8, Aug. 1994.

[3.26] Sullivan D, "Nonlinear FDTD Formulations Using Z Transforms", *IEEE MTT*, vol. 43, no. 1, Mar. 1995, pp. 676–682.

[3.27] Buechler D, et. al. "Modelling Sources in the FDTD Formulation and Their Use in Quantifying Source and Boundary Condition Errors", *IEEE MTT*, vol. 43, no. 4, Apr. 1995, pp. 810–814.

[3.28] Railton C, et. al., "Optimized Absorbing Boundary Conditions for the Analysis of Planar Circuits Using the FDTD Method", *IEEE MTT*, vol. 41, no. 2, Feb. 1993, pp. 290–296.

[3.29] Zhiqiang B, *et. al.*, "A Dispersive Boundary Condition for Microstrip Component Analysis Using the FDTD Method", *IEEE MTT*, vol. 40, no. 4, Apr. 1992, pp. 774–777.

[3.30] Huang T, Houshmand B and Itoh T, "The Implementation of Time-Domain Diakoptics in the FDTD Method", *IEEE MTT*, vol. 42, no. 11, Nov. 1994, pp. 2149–1155.

[3.31] Chen Q and Fusco V, "Three Dimensional FDTD Slotline Analysis on a Limited Memory Personal Computer", *IEEE MTT*, vol. 43, no. 2, Feb. 1995, pp. 358–361.

CHAPTER 4

Microstrip Antennas

4.1 Introduction

Chapter 3 discussed the application of the 3D FDTD method to the simulation of electromagnetic wave propagation. This chapter discusses some background theory on the microstrip antennas which are simulated using the 3D FDTD method in Chapter 7.

Microstrip is a substrate which guides high frequency signals and, in many applications, has replaced waveguides. It has a dielectric substrate mounted onto a ground plane, with a copper track etched on the substrate. The simplest form of a microstrip antenna is a rectangular patch fed from underneath the patch or from a copper feed.

A microstrip antenna is like a resonant cavity with a high Q factor. The high Q factor has the disadvantage that it leads to a small bandwidth (typically only a few per cent) and that its resonance leads to increased simulation times for time-domain methods. Chapters 3, 6 and 7 discuss techniques which reduce this problem and methods to improve accuracy and to reduce run-times. These methods include sub-gridding around discontinuities and parallel processing.

4.2 Microstrip Antenna Construction

A microstrip antenna is made by etching a copper track on a dielectric substrate. The pattern produced defines the resonant frequencies and the radiation pattern of the antenna. Figure 4.1 shows an example of a microstrip antenna patch. It has an antenna feed which is normally matched to 50Ω. A match is achieved between the antenna head and the line-feed by off-setting the antenna head from the centre of the line-feed by a known distance.

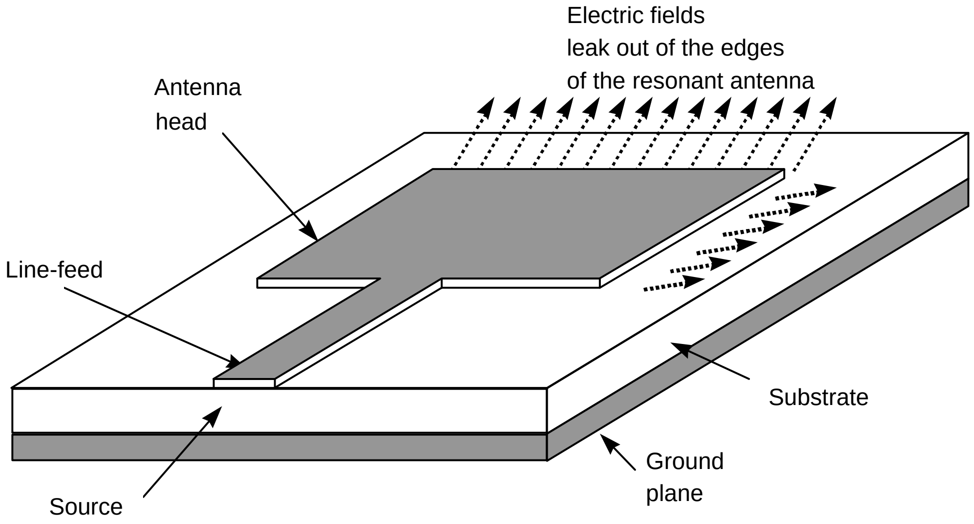


Figure 4.1: Patch Antenna

The applied wave travels into the antenna head and spreads out underneath it. It then reaches the edges of the antenna where some of the energy reflects back and the rest of it radiates out into free-space. The reflected wave then resonates back and forward inside the antenna head until it dies away. Some of this resonant energy returns to the source, some is dissipated in the substrate and the rest of it is radiated out into free-space.

If the frequency of the wave is at a resonant point then the electric fields around the edges have a maximum amplitude. Thus, the radiated electric fields will be at a maximum at resonant frequencies. Figure 4.2 shows some of the reflections.

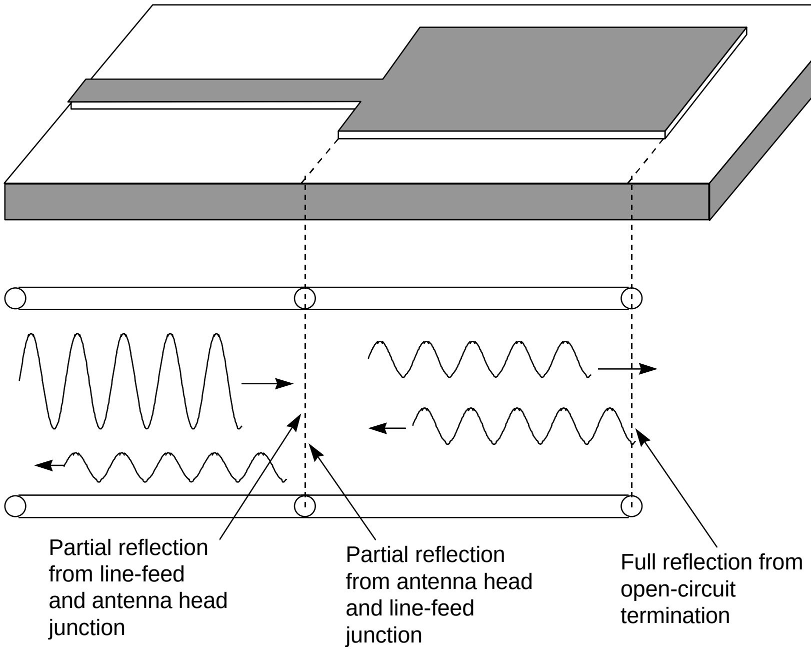

Partial reflection from line-feed and antenna head junction
Partial reflection from antenna head and line-feed junction
Full reflection from open-circuit termination

Figure 4.2: Rectangular antenna patch

4.3 Antenna Substrates

The dielectric constant of the antenna substrate sets the wavelength of the wave within the antenna. Table 4.1 lists some typical substrates. In general, the larger the dielectric constant the smaller the wavelength. For example, an alumina antenna has a smaller patch than an equivalent RT Duriod 5880 antenna because the wavelength in the alumina is almost one-half of that in Duriod. As an approximation the resonant frequency of the antenna occurs when the applied wave has a wavelength which is twice the length of the antenna, that is, the antenna length is half the wavelength of the applied signal.

Table 4.1: Microstrip substrate material

Substrate	*Dielectric constant (e_r)*
RT Duriod 5880	2.1
Polyguide 165	2.32
Fluroglas 600 (PTFE glass cloth)	2.52
RT Duriod 6006 (PTFE)	6.0
Alumina	9.9

4.4 Antenna Modes

Figure 4.3 shows an example of an antenna of length *L* and width *W*. There are three main methods for analysing patch antennas: the transmission-line model; the cavity model; and, the integral equation method.

Patch antennas resonate at multiples of half-wavelength waves, that is, when the applied wavelength is approximately one-half the length of the antenna, one wavelength, three-half wavelengths, and so on. These resonant frequences cause antenna modes.

If the applied electric field has only a *z*-direction component and the magnetic field has only an *x*-component then the wave propagates in the *y*-direction. A transverse magnetic mode (TM) exists when the H_z field is zero and a transverse electric (TE) modes exist when E_z field is zero. Thus, as a microstrip antenna has zero H_z field then it only has TM modes.

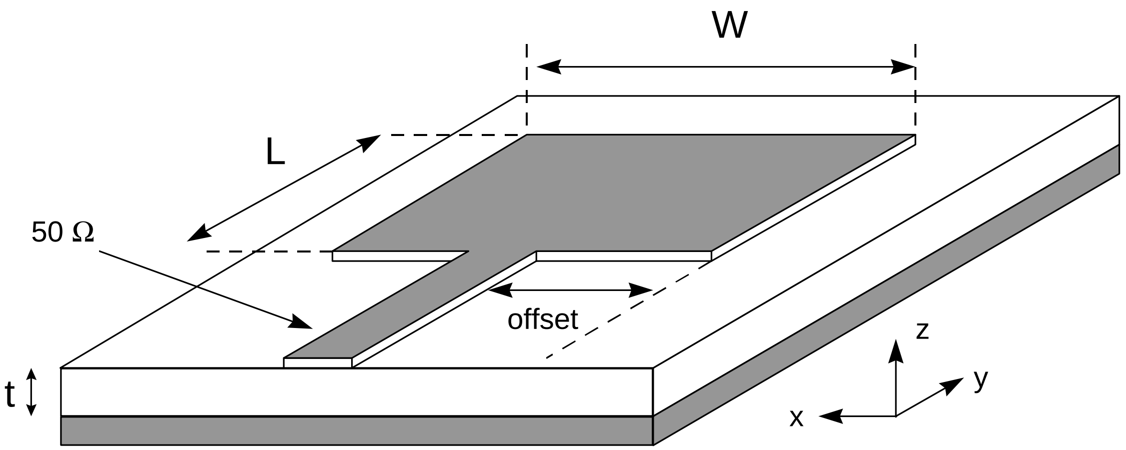


Figure 4.3: Patch Antenna

The electric field at resonance under the patch is given by:

$$E_Z = E_0 \cos\left(\frac{mpx}{W}\right)\cos\left(\frac{npy}{L}\right) \tag{4.1}$$

where *m*, *n* are the modes which are 0, 1, 2, and so on. The *n* value represents resonance across the length of the antenna and *m* the resonance across the width of the antenna.

The resonant frequencies are thus given by:

$$f_{mn} = k_{mn}\frac{c}{2p\sqrt{e_r}} \tag{4.2}$$

where

$$k_{mn}^2 = \left(\frac{mp}{W}\right)^2 + \left(\frac{np}{L}\right)^2 \tag{4.3}$$

Table 4.2 lists the resonant frequencies for modes from TM_{01} up to TM_{33} for a 12.45×16.00 mm^2 antenna with a dielectric of 2.32, using equations (4.1) – (4.3). It can be seen that it resonates at the frequencies of 4.26, 8.52, 12.78 GHz, and so on.

Table 4.2: Antenna modes

n	*m*	k_{mn}	f_{mn} *(GHz)*
0	1	196.35	4.26
0	2	392.7	8.52
0	3	589.05	12.78
1	0	252.34	5.48
1	1	319.73	6.94
1	2	466.78	10.13
2	0	504.67	10.95
2	1	541.52	11.75
2	2	639.46	13.88
3	0	757.01	16.43
3	1	782.06	16.97
3	2	852.81	18.51
3	3	959.19	20.82

Equation (4.2) is a good approximation for the resonant frequencies but it assumes that there are perfect magnetic walls around the patch and thus does not take into account the fringing fields at the edges. James *et al.* [4.2] suggest an improved empirical formula, which is:

$$f_{r1} = f_{r0} \frac{e_r}{\sqrt{e_r(W) e_r(L)}} \frac{1}{1+\Delta} \tag{4.4}$$

where

$$\Delta = \frac{t}{a}\left[0.882 + \frac{0.164(e_r - 1)}{e_r^{\,2}} + \frac{(e_r + 1)}{p e_r} \times \left\{0.758 + \ln\left(\frac{W}{t} + 1.88\right)\right\}\right] \tag{4.5}$$

$$e_e(u) = \frac{e_r + 1}{2} + \frac{e_r - 1}{2}\left[1 + \frac{12t}{u}\right]^{-\frac{1}{2}} \tag{4.6}$$

4.5 Design of Microstrip Antennas

This section discusses the design and analysis of microstrip antennas which are used in Chapter 8 to design and predict antenna performance.

4.5.1 Antenna dimensions

The width and length of a patch can be found by a mixture of analytical analysis and empirical methods. It can be shown from [4.1], [4.2] that the width can be calculated from:

$$W = \frac{c}{2f_r}\left(\frac{e_r + 1}{2}\right)^{-\frac{1}{2}} \tag{4.7}$$

The *c* divided by $2f_r$ term gives one-half a wavelength in free-space and the other term scales the equation to give a half-wavelength in the substrate.

The length is also found by calculating the half-wavelength value and then subtracting a small length to take into account the fringing fields, it is given by:

$$L = \frac{c}{2f_r\sqrt{e_e}} - 2.\Delta l \tag{4.8}$$

where

$$\Delta l = 0.412h\frac{\left(e_e + 0.3\right)\cdot\left(\frac{W}{h} + 0.264\right)}{\left(e_e - 0.258\right)\cdot\left(\frac{W}{h} + 0.8\right)} \tag{4.9}$$

and

$$e_e = \frac{e_r + 1}{2} + \frac{e_r - 1}{2}\left[1 + \frac{12t}{W}\right]^{-\frac{1}{2}} \tag{4.10}$$

Figure 4.4 shows a plot of Equation (4.7) with a dielectric constant of 2.32 and Figure 4.5 shows a plot of Equation (4.8). The first graph shows the variation of the desired antenna width with resonant frequency and the second shows the variation of antenna length with resonant frequency.

4.6 Microstrip Antenna Analysis

After designing the antenna it can be analysed using the methods given next. The most important parameters are the radiation pattern, the input impedance, the bandwidth, the beamwidth and the gain [4.3], [4.4].

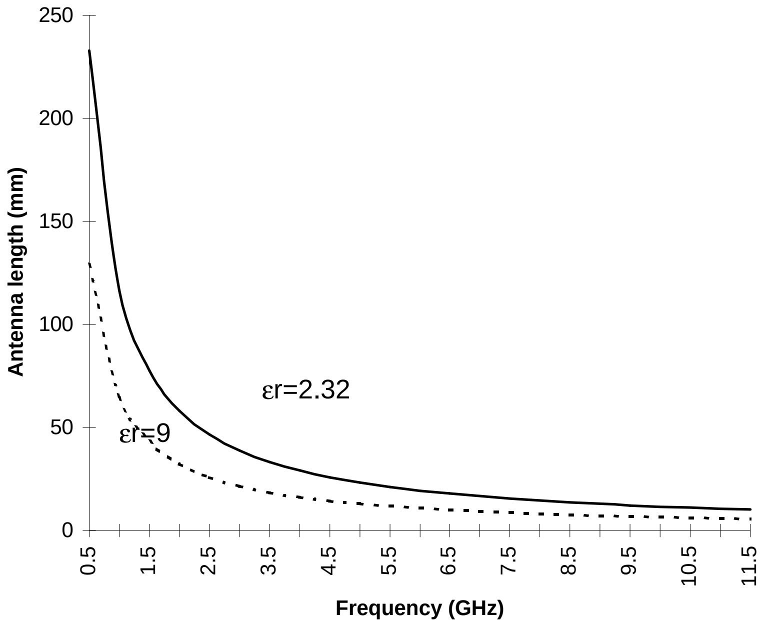


Figure 4.4: Patch antenna width

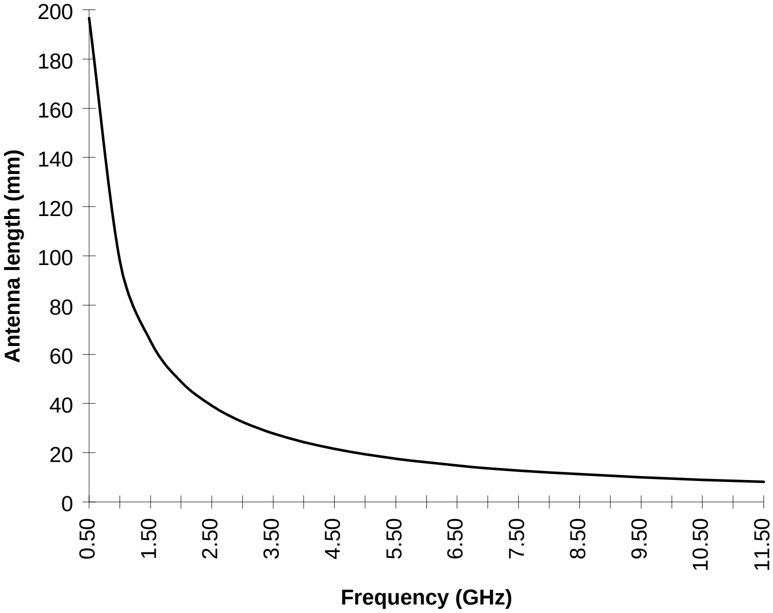


Figure 4.5: Patch antenna length

4.6.1 Radiation pattern

The radiation pattern can be predicted using a simple transmission line model. Equations (4.11) and (4.12) define the patterns.

$$F(\theta) = \frac{\sin\left(\frac{k_0 W}{2}\cos\theta\right)}{\frac{k_0 W}{2}\cos\theta}\sin\theta \tag{4.11}$$

$$F(\phi) = \frac{\sin\left(\frac{k_0 h}{2}\cos\phi\right)}{\frac{k_0 h}{2}\cos\phi}\cos\left(\frac{k_0 L}{2}\cos\phi\right) \tag{4.12}$$

where $k_0 = \frac{c}{l}$. Figure 4.6 shows a plot of Equations (4.11) and (4.12) for an antenna of 58.21×48.9 mm^2 at a resonant frequency of 2 GHz.

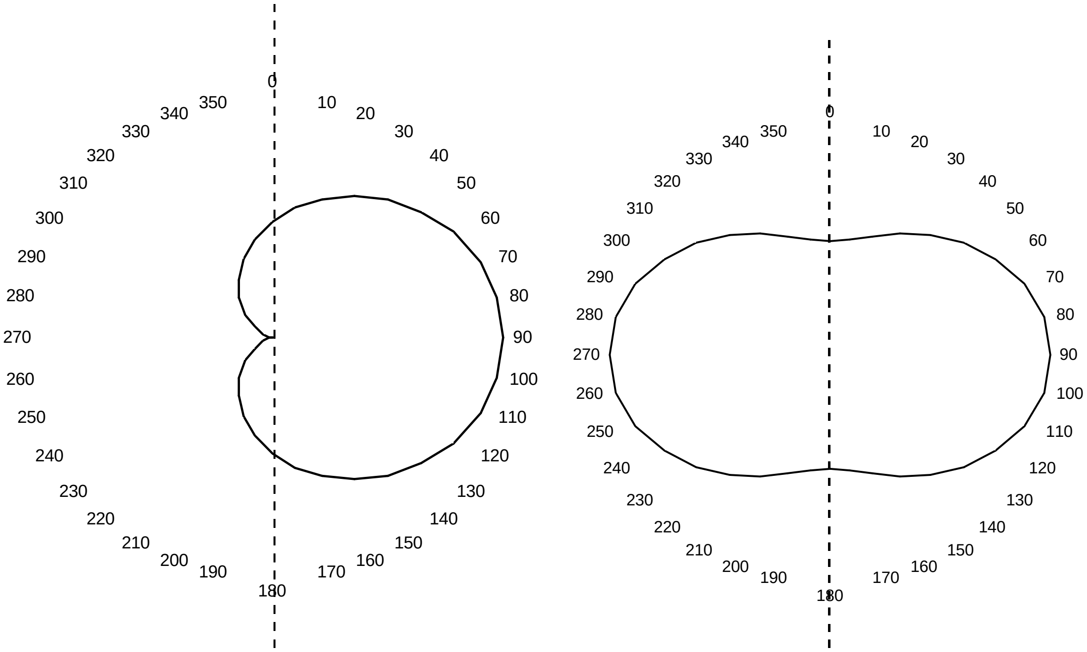


Figure 4.6: F(θ) and F(ϕ) field patterns

4.7 References

[4.1] Bahl IJ, Bhartia P, "Microstrip Antennas", *Artech House*, 1980.

[4.2] James JR & Hall PS, "Handbook of Microstrip Antennas", *IEE Electromagnetic Waves Series*, No. 28, Peter Peregrinus, 1989. Vol. 1.

[4.3] James JR & Hall PS, "Handbook of Microstrip Antennas", *IEE Electromagnetic Waves Series*, No. 28, Peter Peregrinus, 1989. Vol. 2.

[4.4] James JR, Hall PS and Wood C, "Microstrip antennas, theory and design", *IEE Electromagnetic Waves Series*, No. 19, Peter Peregrinus, 1989.

CHAPTER 5

Parallel Processing of 3D FDTD Method using Transputers

5.1 Introduction

Computer systems have generally evolved around a single centralised processor with an associated area of memory. This main processor performs most of the operations within the computer and also controls reads and writes to and from memory. This type of arrangement is useful in that there is little chance of a conflict when addressing any peripheral as only the single processor can access it. With the evolution of microelectronics it is now possible to build computers with many processors. It is typical on modern computers to have several processors, apart from the central processor. For example many computers now have dedicated processors to control the graphical display, processors to controls input/ output functions of the computer, processors to control the hard-disk drive, and so on.

Computer systems are also now being designed with several processors that run application programs. Each of these processors can access their own localised memory and/or a shared memory. This type of multi-processor system, though, leads to several problems, including device conflicts and processor synchronisation. Figure 5.1 illustrates the two types of system.

A memory conflict occurs when a process tries to read from or write to an area of memory at the same time as another is trying to access it. Normally, multi-processor systems have mechanisms that lock areas of memory when a processor is accessing it.

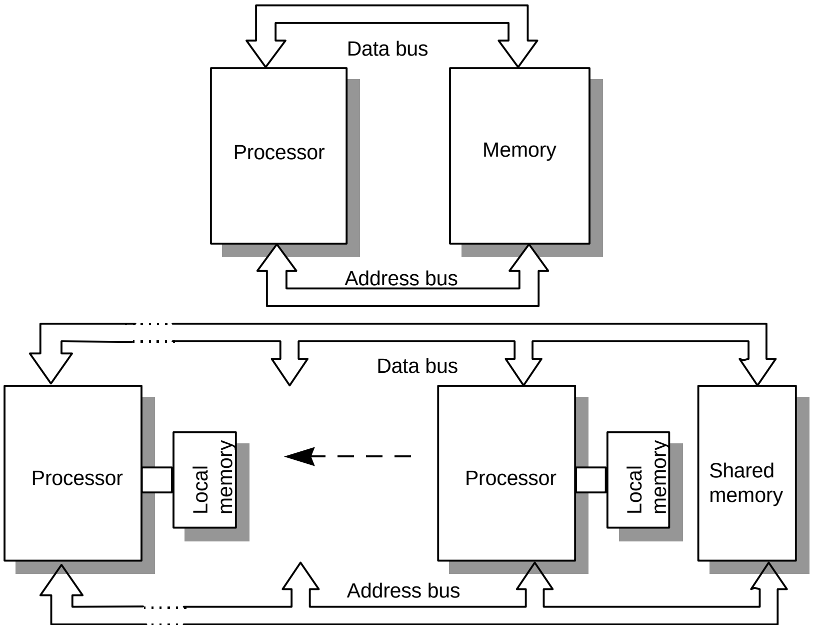


Figure 5.1: Single and multi-processor systems

Parallel systems require processor synchronisation because one or more processors may require data from other processors. This synchronisation can either be hard-wired into the system using data and addressing busses, or by a master controlling processor that handles the communication among slave processors (processor farms). They may also be controlled by the operating system software.

This and the next chapter discuss two types of multi-processing, one using transputer arrays and the other using workstations connected over a local area network (LAN).

5.2 Background

Several researchers are now investigating the application of parallel processing to the FDTD method. These include Fusco [5.1] and [5.2], Excell and Tinniswood [5.3], [5.4], Buchanan and Gupta [5.5]-[5.9], and Pala [5.10]. Fusco, at Queen's University of Belfast, used small transputer arrays to implement a parallel 2D FDTD problem based on diakoptics. With this method Fusco replaces some of the finite difference equations by resistive analogues.

Excell and Tinniswood, at the University of Bradford, have applied the method to the simulation of electromagnetic waves on human tissues. They are currently involved in the Parallel Electromagnetic Programming Support Environment (PEPSE) which is part of the ESPRIT EUROPORT program. The main aim of this project is to demonstrate scaleable and portable parallel implementations of industry standard programs. The parallelisation used can either use massively parallel processors (MPP) or clustered workstations.

5.3 Parallel Techniques

5.3.1 Pipelines and parallel streams

There are two main methods used when dividing computational tasks to individual processors. Either computations are divided into stages in a pipeline or they are divided into parallel streams, as illustrated in Figure 5.2. A mixed method uses a mixture of pipelines and parallel streams.

The pipeline method is preferable when there is a large number of computations on a small amount of data. Distributing data between streams can be awkward, since calculations often involve two or more consecutive items of data. Parallel streams are preferable for simple operations on large amounts of data, which is the case in the 3D FDTD method.

A major problem with pipelines is that it is difficult to ensure that all the processors have an equal loading. If one processor has a heavier work-load than its neighbours then this processor holds-up the neighbours while they are waiting for data from the burdened processor.

It is always important to recognise the inherent parallelism in the problem and wether to allocate fast processors to critical parts and slower ones for the rest, or to equalise the workload, called load balancing. The 3D FDTD method is relatively easy to load balance as, in most large problems, each processor performs the same calculation on the same amount of data.

5.3.2 Processor Farms

Processor farming is a technique for distributing work with automatic load balancing. It uses a master processor to distribute tasks to a network of slaves. The slave processors only get tasks when they are idle.

It is important in a parallel system that processor tasks are large enough because each task has its overheads. These include the handling overhead of the master controller and also the inter-procesor communication. If the tasks are too small then these overheads take a significant amount of time and cause bottlenecks in the system [5.10].

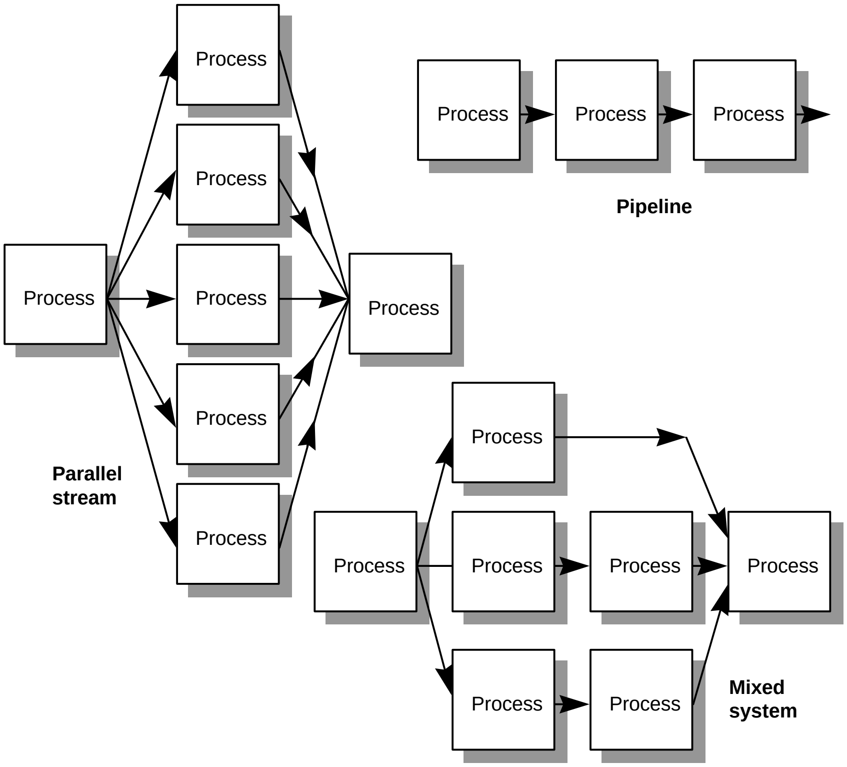


Figure 5.2: Pipeline, parallel stream and mixed systems

5.4 Transputer Simulations

5.4.1 Transputers

A transputer is a device developed in the UK by INMOS Limited. They are mounted onto a daughter board that fits into a standard PC or workstation. Within each transputer there is a powerful microprocessor, several communication ports, timers, clocks, and so on. Figure 5.3 illustrates the basic architecture of a single device.

They can be used to execute an application program as a single process on one transputer or with other transputers to form a large array in which each transputer communicates with its neighbour by means of point-to-point communications (as illustrated in Figure 5.4).

A typical transputer has a 32-bit RISC (reduced instruction set code) processor, on-board and local memory, full 64-bit floating-point processing and a high speed serial link to communicate with its neighbouring transputers. Each transputer is thus equivalent to a powerful microcomputer. The T8*xx* series process data at 30 Mips (million operations per seconds) or 4.3 Mflops (million floating point operations per second). It communicates with other devices using either a coaxial or fibre optic cable at rates of 1, 5, 10 or 20 Mbps.

The T8*xx* series of transputers are 32-bit RISC processors with a floating point unit (the T4*xx* series transputers have the same processors but have no floating point unit). Newer, faster processors, named the T9000s, are now available giving improvements in the communications and processor performance.

The transputer is well suited to parallel problems and is relatively inexpensive to buy. One of its major advantages is that it allows scaleable parallel designs.

They can communicate and process at the same time, and can thus act as both slaves and routing devices at the same time. These slaves can be arranged into a pipeline or any other convenient network.

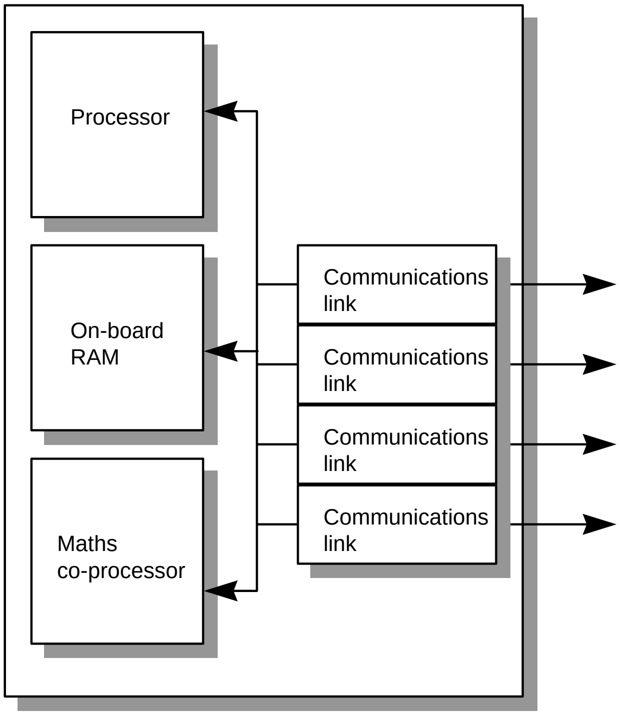


Figure 5.3: A transputer

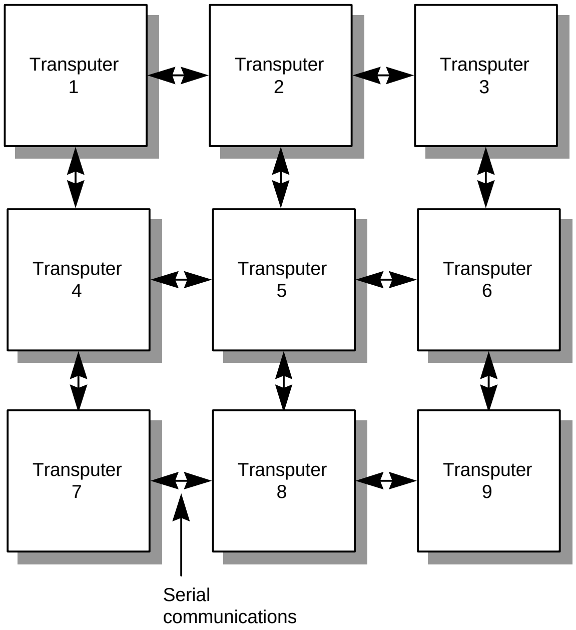


Figure 5.4: Transputer array

5.4.2 Communications links

When a processor wishes to send data to its neighbour it must wait until it is ready. When both are ready, the data is sent and both processors can continue processing their data. Each link can input and output data at the same time and since the links are autonomous they can be working while the processor is doing something else. This is an important factor in the transputers performance.

If the transputer wishes to communicate down a link, the transputer gives control of the process to the link. The link then handles the communication while the processor either waits for the data or finds another process to run. Data is then sent serially in byte packets and each byte is acknowledged. The link always tries to communicate and if there is no response, it assumes that the process at the other end is not ready. It then waits indefinitely until a response is received. If the response is received then the link tries to communicate the next byte, and so on until it completes the transmission.

The link sends single bytes of data between the processors wrapped within a packet of asynchronous data. This data begins with the bit sequence ‘11’ and ends with a stop bit of ‘0’. Thus, 11 bits are used to send one byte of data, if the link bit rate is 5 Mbps then the maximum data rate will be:

$$\text{Maximum data rate} = \frac{5\ \text{Mbps}}{11\ \text{bits per byte}} = \ 454\ 545\ \text{B / sec}$$

When the receiver gets data it sends an acknowledgement packet that has a bit sequence of ‘10’. On average, when transmitting in both directions, and taking into account acknowledgements and delays, it takes 2 bytes to transmit a single byte of data. Thus the data rate for data for each direction will be:

$$\text{Maximum data rate (in each direction)} = \frac{5\ \text{Mbps}}{16\ \text{bits per byte}} = 312\,500\ \text{B / sec}$$

5.4.3 Simulation using the 3D FDTD method

To simulate the structure in Figure 5.5, first the problem is segmented into physical domains. To even the load on each transputer, each is assigned roughly the same amount of cells. Then each transputer operates on one of the domains.

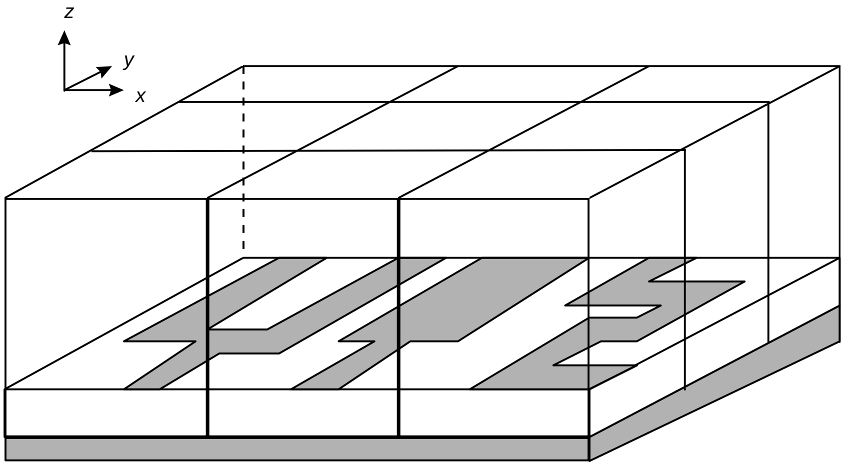


Figure 5.5: Segmentation of problem with a 3×3 transputer array

In the 3D FDTD method the present calculation depends only on the previous time-step and no cell has to wait until its neighbours have completed their calculations. An interchange of data then occurs at the end of each time-step. This enables a high degree of parallelism as there is no sequential sweeping over the problem. The only limitation on this parallelism is the reporting of boundary conditions to each of the neighbouring physical domains.

It is important that each transputer handles an equal share of cells as this balances the burden on each and reduces the time the transputers are waiting for boundary conditions to be communicated.

The number of cells that each transputer operates on will thus be:

$$\text{Number of nodes per domain} = \frac{NX \times NY \times NZ}{NT} \qquad (5.1)$$

where NX, NY and NZ are the number of cells in the x-, y- and z-directions, respectively, and NT is the number of transputers.

After each time iteration the boundary conditions are transmitted at either 1, 5, 10 or 20 Mbps to neighbouring physical domains which may result in a time large overhead. Thus, for a given size of problem, more parallel processes lead to a reduction in processing time but increases the time spent with the inter-domain communications. Figure 5.6 shows the number of interfaces for a given number of domains and Figure 5.7 gives an example of a 32 domain problem.

The number of interfaces in which boundary conditions are transmitted depends on the number of domains. A 2 domain problem has only one interface, a four domain has 4, an nine domain has 12, and so on. Table 5.1 gives examples of the number of interfaces.

The number of boundary conditions transmitted will be twice the number of interfaces as data has to be transmitted in both directions.

Table 5.1: Domain problem

Transputer array (NT×NT)	1×1	2×2	3×3	4×4	5×5	6×6	7×7
Number of domains	1	4	9	16	25	36	49
Interfaces	0	3	12	24	40	60	84
Boundary conditions transmitted	0	6	24	48	80	120	168

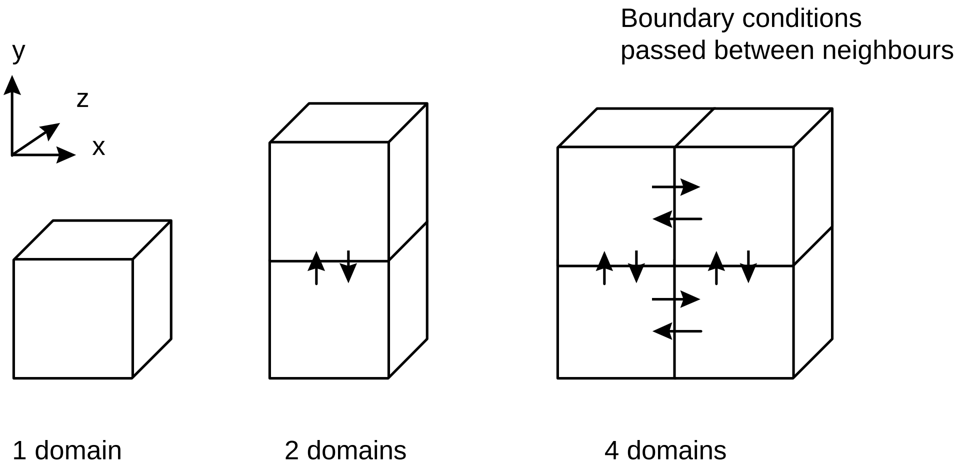


Figure 5.6: Boundary conditions passed between domains

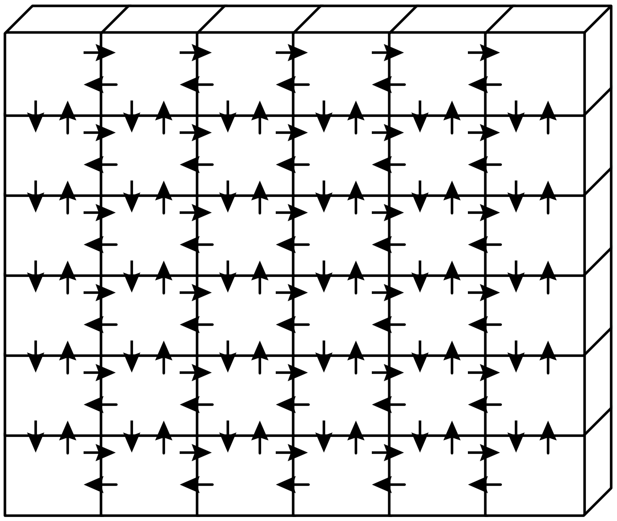


Figure 5.7: 32 domain set-up

In general, for an $NT \times NT$ transputer array, the number of interfaces will be:

$$Interfaces = 2 \times (NT) \times (NT - 1) \tag{5.2}$$

The number of cells transmitted depends on the total number of cells at all the interfaces. Figure 5.8 shows an example 5×4 transputer array. In this case, the number of cells transmitted in the *x*-direction will be:

$$4\times 4\times \frac{NY}{4}\times NZ \qquad (5.3)$$

and in the *y*-direction the number of cells transmitted will be:

$$3\times 5\times \frac{NX}{5}\times NZ \qquad (5.4)$$

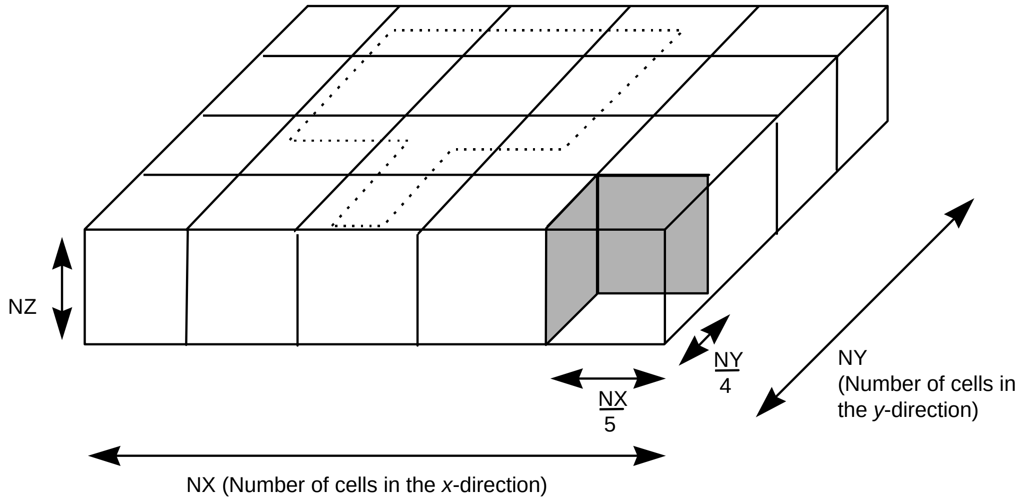

NZ
NY/4
NX/5
NY
(Number of cells in the *y*-direction)
NX (Number of cells in the *x*-direction)

Figure 5.8: 5×4 array

Thus, the total number of cells transmitted will be summation of the two. In general the number of cells transmitted for an NT_X×NT_Y array will be:

$$\text{Cells transmitted} = \left(\frac{NY}{NT_Y}\right)\times NZ\times\left(NT_X-1\right)\times\left(NT_Y\right)+\left(\frac{NX}{NT_X}\right)\times NZ\times\left(NT_X\right)\times\left(NT_Y-1\right)$$

(5.5)

Each cell transmits data in both directions and there are six field components for each cell (E_x, E_y, E_z, H_x, H_y and H_z). As was discussed in Section 5.3.1 it takes an average of 16 bits to transmit a single byte of data (in both directions). Thus, the number of bits transmitted will be Equation (5.5) multiplied by 16 and then multiplied by 6 to take into account the 6 field components per cell. Thus the total time to transmit all the cells depends on the bit rate of the link, and will be given by:

$$\text{Total time taken to transmit cells} = \frac{16 \times \text{Cells transmitted} \times 6}{\text{Transmission bit rate}} \quad \text{s} \tag{5.6}$$

Each processor operates on a NX / NT_X by NY / NT_Y by NZ domain size, then the total time to process the problem can be approximated by:

$$\text{Processor time taken} = \frac{NX \times NY \times NZ \times \text{Time per iteration}}{NT_X \times NT_Y} \quad \text{s} \tag{5.7}$$

The total simulation will thus be the summation of Equation (5.6) and (5.7).

Table 5.2 and Figure 5.9 shows the total time taken against a processor array size for a 100×100×20 grid for 5 000 time iterations. These simulations are based on a 1 Mbps inter-communication rate and assume 4 bytes per floating point value. It can be seen that the optimum number of transputers, for this inter-communication rate, is around 2×2 or 3×3. It can also be observed that if more than 9 transputers are used, there is an increase in the simulation time.

Table 5.3 and Figure 5.10 show the expected simulation time for 5 Mbps, and Table 5.4 and Figure 5.11 show the simulation time for 10 Mbps.

It can be seen from Table 5.3 that an array of 3×3 and 4×4 significantly reduces the simulation time, from 30 000 seconds for a single processor down to 4 101 and 3 027 seconds, respectively. The communication overhead is also less than 40 %. With a 5×5 array (25 processors) there is little significant decrease in simulation time. This is due to 56.1 % overhead in link communications. Table 5.3 also shows that above a 5×5 array the simulation time actually increases. Similar conclusions can be drawn for Figures 5.10 and 5.11.

Table 5.2: Computation time for a 100×100×20 array with 5 000 iterations at 1 Mbps

Processor array	Total communications time (s)	Total processing time (s)	Total simulation time (s)	Transmission overhead (%)
1×1	0	30 000	30 000	0.0
2×2	1 920	7 500	9 420	20.4
3×3	3 840	3 333	7 173	53.5
4×4	5760	1 875	7 635	75.4
5×5	7 680	1 200	8 880	86.5
6×6	9 600	833	10 433	92.0
7×7	11 520	612	12 132	95.0
8×8	13 440	469	13 909	96.6
9×9	15 360	370	15 730	97.6
10×10	17 280	300	17 580	98.3

Table 5.3: Computation time for a 100×100×20 array with 5 000 iterations at 5 Mbps

Processor array	Total communications time (s)	Total processing time (s)	Total simulation time (s)	Transmission overhead (%)
1×1	0	30 000	30 000	0.0
2×2	384	7 500	7 884	4.9
3×3	768	3 333	4 101	18.7
4×4	1 152	1 875	3 027	38.1
5×5	1 536	1 200	2 736	56.1
6×6	1 920	833	2 753	69.7
7×7	2 304	612	2 916	79.0
8×8	2 688	469	3 157	85.2
9×9	3 072	370	3 442	89.2
10×10	3 456	300	3 756	92.0

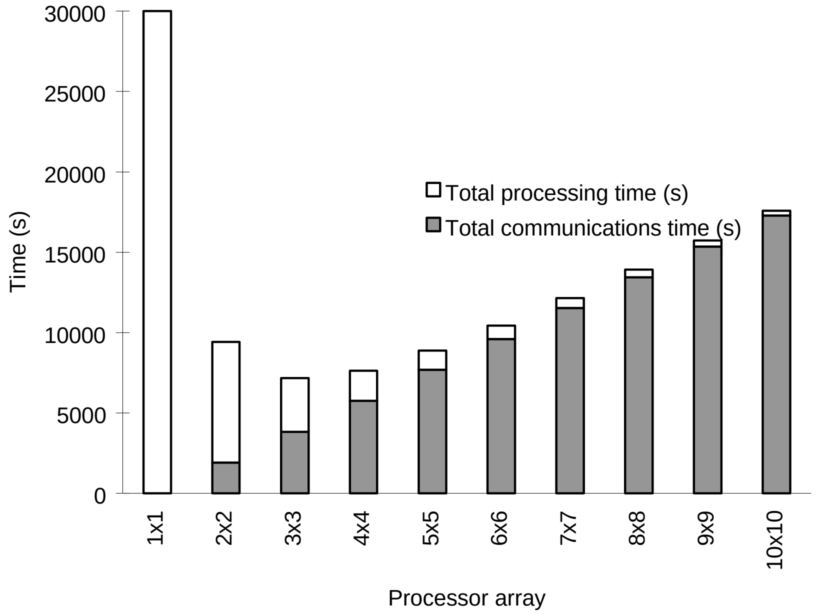


Figure 5.9: Computation time for a 100×100×20 array with 5 000 iterations at 1 Mbps

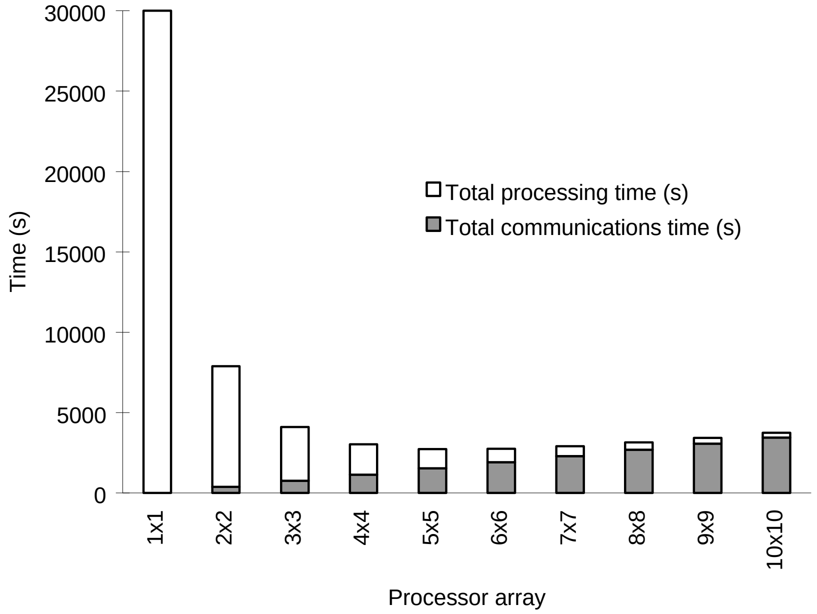


Figure 5.10: Computation time for a 100×100×20 array with 5 000 iterations at 5 Mbps

Table 5.4: Computation time for a 100×100×20 array with 5 000 iterations at 10 Mbps

Processor array	Total communications time (s)	Total processing time (s)	Total simulation time (s)	Transmission overhead (%)
1×1	0	30 000	30 000	0.0
2×2	192	7 500	7 692	2.5
3×3	384	3 333	3 717	10.3
4×4	576	1 875	2 451	23.5
5×5	768	1 200	1 968	39.0
6×6	960	833	1 793	53.5
7×7	1 152	612	1 764	65.3
8×8	1 344	469	1 813	74.1
9×9	1 536	370	1 906	80.6
10×10	1 728	300	2 028	85.2

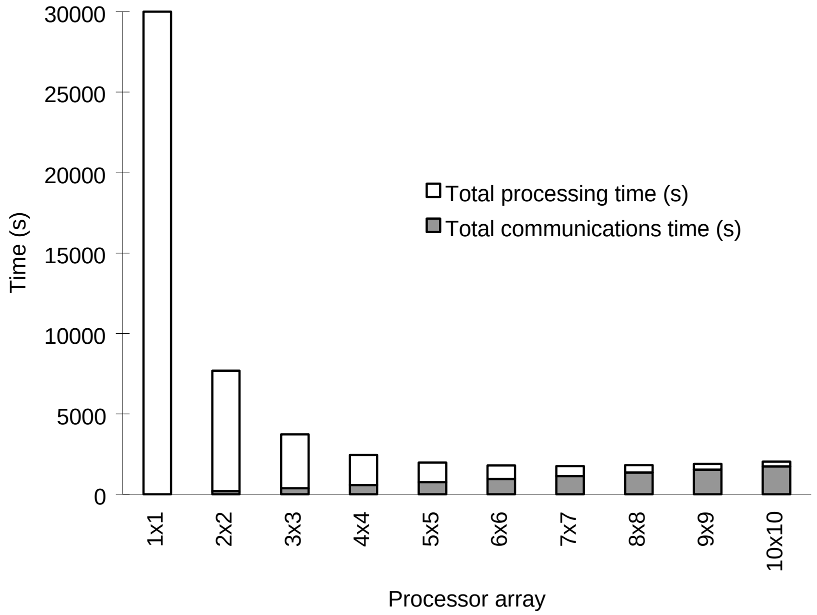


Figure 5.11: Computation time for a 100×100×20 array with 5 000 iterations at 10 Mbps

Table 5.5 summarised the communication link overhead for various bit rates. It can be seen that the communication link overhead significantly reduces with increasing bit rates. Table 5.6 summarises the total simulation time and Figure 5.12 shows the variation of the simulation time with various link bit rates.

Table 5.5: Communications overhead for link bit rates

Processor array	Overhead (%), 1 Mbps	Overhead (%), 5 Mbps	Overhead (%), 10 Mbps	Overhead (%), 20 Mbps
1×1	0.0	0.0	0.0	0
2×2	20.4	4.9	2.5	1.3
3×3	53.5	18.7	10.3	5.4
4×4	75.4	38.1	23.5	13.3
5×5	86.5	56.1	39.0	24.2
6×6	92.0	69.7	53.5	36.5
7×7	95.0	79.0	65.3	48.5
8×8	96.6	85.2	74.1	58.9
9×9	97.6	89.2	80.6	67.5
10×10	98.3	92.0	85.2	74.2

Table 5.6: Simulation times for link bit rates

Processor array	Total simulation time (s), 1 Mbps	Total simulation time (s), 5 Mbps	Total simulation time (s), 10 Mbps	Total simulation time (s), 20 Mbps
1×1	30 000	30 000	30 000	30 000
2×2	9 420	7 884	7 692	7 596
3×3	7 173	4 101	3 717	3 525
4×4	7 635	3 027	2 451	2 163
5×5	8 880	2 736	1 968	1 584
6×6	10 433	2 753	1 793	1 313
7×7	12 132	2 916	1 764	1 188
8×8	13 909	3 157	1 813	1 141
9×9	15 730	3 442	1 906	1 138
10×10	17 580	3 756	2 028	1 164

5.4.4 Transputer array results

Simulations were conducted for a 100×100×20 array for 5 000 iterations with a 2×2 and a 3×3 array. The total simulation time agreed well with the expected simulation time, with a maximum error of 5 %. This error was probably due to synchronisation problems.

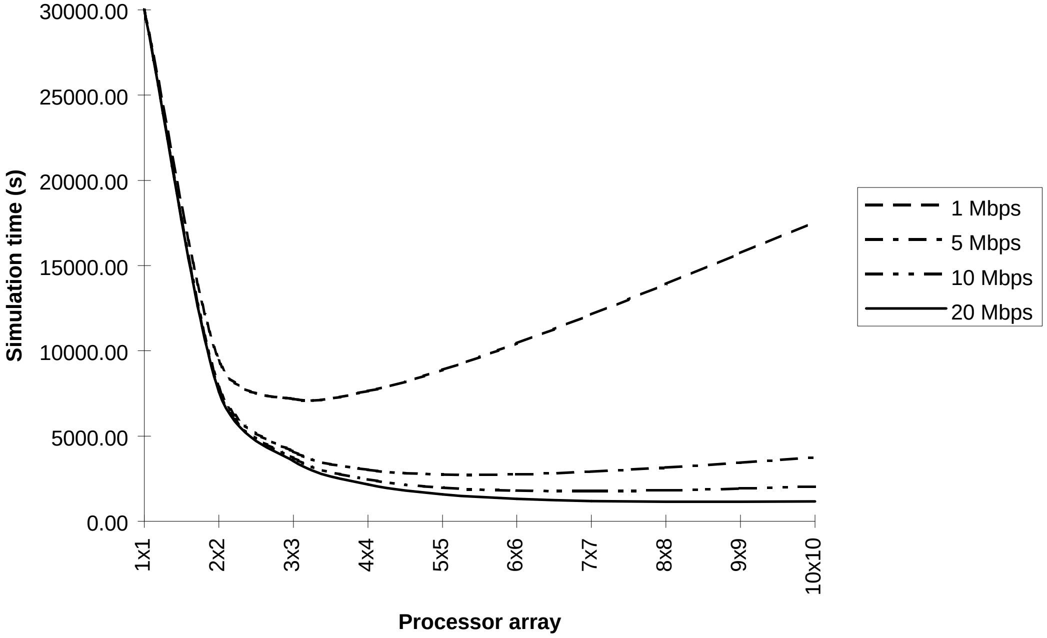
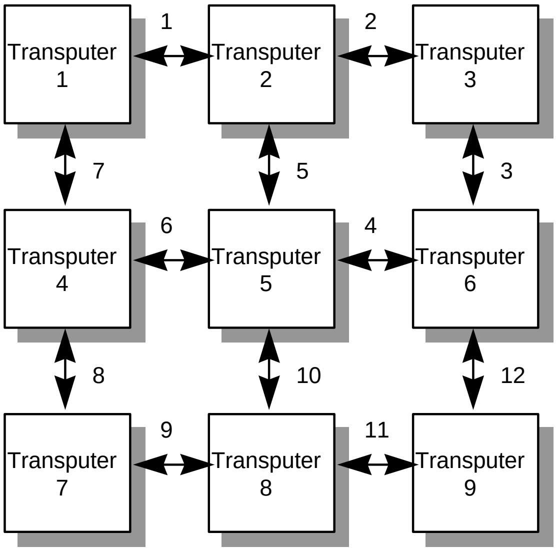


Figure 5.12: Total simulation times for link bit rates

5.5 Improved Parallelisation Method

The simulation in the previous section does not make full use of the inherent, parallelism of the transputer-to-transputer communications. Figure 5.13 shows how transputer communications are conducted with a round robin technique, that is, transputer 1 passes its data to transputer 2, which then passes its data to transputer 3, and so on. Thus, transputer 2 must wait for transputer 1 before it can send its data to transputer 3. The other transputers in the array must also wait until they receive data from the transputer directly before them in the sequence. This is inefficient in communication time as several transputers could communicate at a time without synchronisation problems. An improved method, for a 3×3 array, is shown in Figure 5.14. The sequence is:

- transputers 1, 4 and 7 transmit their boundary data to transputers 2, 5 and 8, respectively;
- upon receipt of the data, transputers 2, 5 and 8 then transmit their data to transputers 3, 6 and 9, respectively.
- transputers 1, 2 and 3 wait for transputers 4, 5 and 6 to complete their transmission, and then transmit to them;
- transputers 4, 5 and 6 then transmit to transputers 7, 8 and 9, respectively.



Sequence of communications: 1-2-3-4 … 11-12

Figure 5.13: Round robin communications between transputers

This sequence of operations only takes 4 steps as opposed to 12 with the round-robin technique. Table 5.7 summarises the improvement in the number of steps. It can be seen that only 4 or 6 synchronisation steps are required. Figure 5.14 gives examples of a 4×4 array and a 5×5 array.

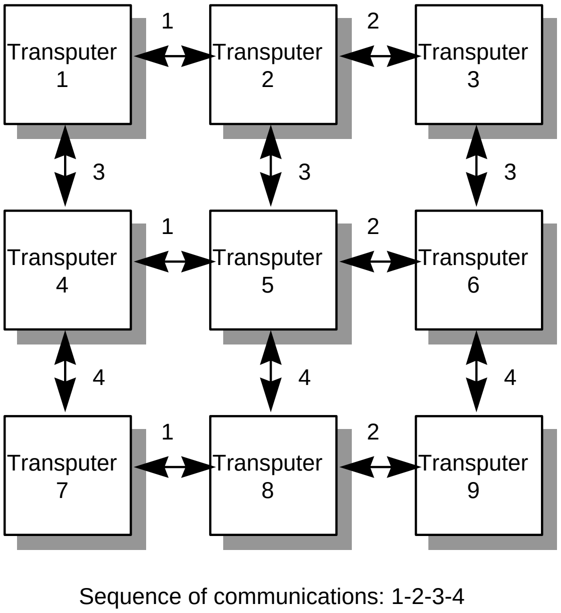


Figure 5.14: Synchronised steps method of communication

Table 5.7: Communication steps

Processor array	Round robin method	Synchronised steps method
1×1	0	0
2×2	4	2
3×3	12	4
4×4	24	4
5×5	40	6
6×6	60	6
7×7	84	6
8×8	112	4
9×9	144	4
10×10	180	6

5.6 Conclusions

This chapter has shown how transputer arrays can be used to simulate the 3D FDTD problem. They are efficient in their parallelism but suffer from a communication overhead. This overhead can be significantly improved if the transputers cells are synchronised so that several transputers communicate at the same time. Higher link bit rates also significantly reduce transmission overheads.

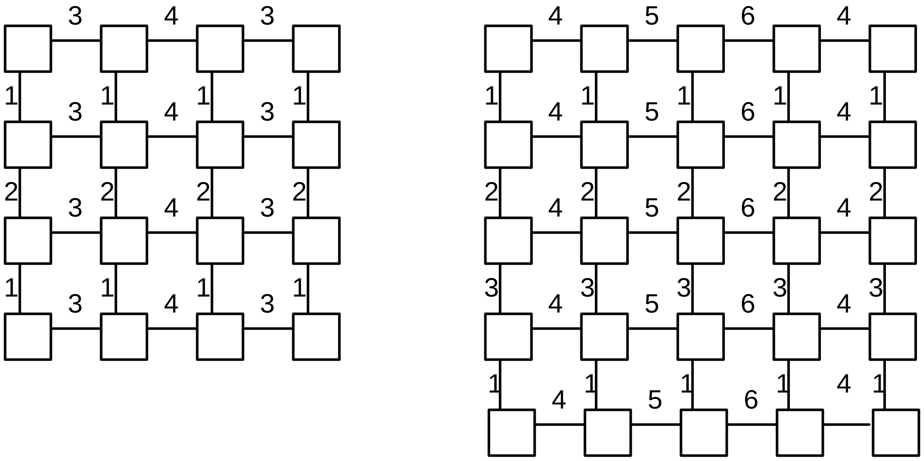


Figure 5.15: Synchronisation steps for a 4×4 array and a 5×5 array

5.7 References

[5.1] Merugu L and Fusco V, “Concurrent Network Diakoptics for Electromagnetic Field Problems”, *IEEE MTT*, vol. 41, no. 4, Apr. 1993, pp. 708–716.

[5.2] Fusco V, Merugu L and McDowall, “An Efficient Diakoptics-based Algorithm for Electromagnetic Field Mapping”, *IEE’s 1st International Conference in Electromagnetics*, Savoy Place, London, Apr. 1991.

[5.3] Excell PS and Tinniswood AD, “A FDTD Program for Parallel Computers”, *QMW 1995 Antenna Symposium*, Queen Mary and Westfield College, July 1995.

[5.4] Excell PS and Tinniswood AD, “Parallel Computation of Large-scale FDTD problems”, *IEE 3rd International Conference in Electromagnetics*, University of Bath, Apr. 1996.

[5.5] WJ Buchanan, NK Gupta “Parallel Processing of the Three-Dimensional Finite-Difference Time-Domain Method”, *National Radio Science Colloquium*, University of Bradford, 7-8 Jul. 1992.

[5.6] WJ Buchanan, NK Gupta, “Simulation of Electromagnetic Pulse Propagation in Three-

Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques”, *Electrosoft ‘93*, Jul. 1993, Southampton.

[5.7] WJ Buchanan, NK Gupta, “Parallel Processing Techniques in EMP Propagation using 3D Finite-Difference Time-Domain (FDTD) Method”, *Journal of Advances in Engineering Software*, vol. 18, 3, 1993.

[5.8] WJ Buchanan, NK Gupta, “Prediction of Electric Fields in and around PCBs – 3D Finite-Difference Time-Domain Approach with Parallel Processing”, *Journal of Advances in Engineering Software*, Dec. 1995.

[5.9] Pala WP, “Parallel FDTD calculation”, *IEE’s 1st International Conference in Electromagnetics*, Savoy Place, London, Apr. 1991.

[5.10] J Hinton & A Pinder, “Transputer Hardware and System Design”, Prentice Hall, 1993.

CHAPTER 6

Parallel Processing of 3D FDTD Method using a Local Area Network (LAN)

6.1 Introduction

Chapter 5 discussed the application of transputer arrays to the simulation of the 3D FDTD method. These devices are well suited to parallel problems but their availability, expense and the limited range of development tools are a limiting factor when building large arrays. They can also be costly when upgrading as, in the case of 3D FDTD simulations, all the processors in the array require to be upgraded.

An alternative parallel processing method is to divide computational tasks amongst networked computers, such as PCs or workstations. As with transputer arrays, each computer is allocated a physical domain for which they compute the electric and magnetic fields. At the end of each time-step, they communicate with other computers with a neighbouring domain. This communication occurs over the LAN connection. This chapter discusses parallel simulations over an Ethernet network.

6.2 Background

Other researchers, such as Excell and Tinniswood [6.1] have applied parallel processing to the 3D FDTD method using a Meiko CS-2 MPP (massively-parallel processors) computer and the KSR-1 ‘virtual shared computer’. These computers are specially designed for parallel processing and have high-speed data links between processors. This chapter discusses the implementation of the method using clustered standard workstations.

6.3 Ethernet

The Xerox Corporation, in conjunction with DEC and Intel, developed the Ethernet network. Standards have since been developed by the IEEE 802 committee. It uses a bus network topology where all nodes share a common bus and only one node can communicate at a time, as illustrated in Figure 6.1. Data frames are transmitted at 10 Mbps and contain both the source and destination addresses. Each node on the network monitors the bus and copies any frames addressed to itself.

6.3.1 Ethernet frame

The data transmitted over the network is wrapped with a frame, as illustrated in Figure 6.2. This frame contains 6 bytes for each of the source and destination addresses (48 bits each), 4 bytes for the frame check sequence (32 bits), 2 bytes for the logical link control (LLC) field length (16 bits) and up to 1518 bytes for the LLC field. Preamble and delay components define the start and end of the frame. Initial preamble and the start delimiter are, in total, 8 bytes long and the delay component is a minimum of 96 bytes long.

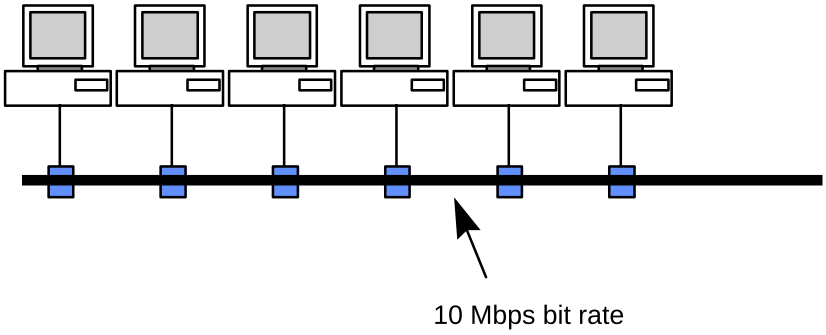


Figure 6.1: Ethernet networks

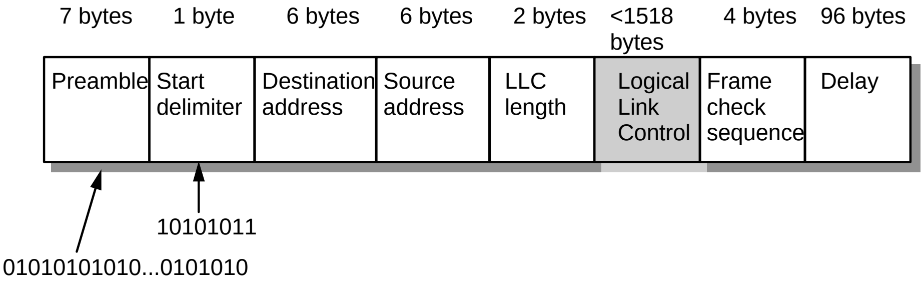


Figure 6.2: Ethernet frame format

The 7-byte preamble that precedes the Ethernet frame has a fixed binary pattern of `10101010..1010` and is used by all nodes on the network to synchronise their clocks and transmission timings. It also informs nodes that a frame is to be sent and for them to check the destination address in the frame. The start delimiter follows the preamble and is a single byte of `10101011`.

A 96-byte period ends the frame and provides the minimum delay between two frames. This slot-time delay allows for the worst-case network propagation delay.

The source and destination addresses contain a 48-bit media access control (MAC) addresses for the sending node and destination address.

The logical link field can contain up to 1518 bytes of information and has a minimum of 46 bytes. Typically, 4 bytes of this field contains control information and the rest is data [6.2]. If the amount of data is greater than the upper limit then multiple frames are sent. Also, if the field is less than the lower limit then it is padded by extra redundant bits.

The frame check sequence (or FCS) is an error detection scheme that is used to determine transmission errors. It is often referred to as a cyclic redundancy check (CRC) or simply as a checksum.

6.3.2 Ethernet frame overhead

An Ethernet frame contains up to 1514 bytes of data. The start and end of the frame and the delay between frames adds another 7+1+6+6+2+4+96 bytes (122 bytes). The effect this overhead has on the data depends on the amount of data sent within the frame. For example, when sending 50 bytes, the overhead is over 200%, but for 1514 bytes it is only 8%.

6.4 FDTD model simulation

It is important when dividing the processing tasks to ensure that each processor has a relatively large task because of the inter-processor communications overhead. Another important directive is that the segmentation of a problem should also be relatively simple to set-up and the processor array should be scaleable [6.1], that is, it should be relatively easy to scale the problem from an $n\times n$ to $m\times m$ array.

As with transputers arrays the most efficient segmentation of the problem occurs when the *x*- and *y*-directions are segmented and the *z*-direction is not. This is because the *x*- and *y*-direction requires a larger grid than the *z*-direction, as illustrated in Figure 6.2. The *x*- and *y*-direction grid typically have at least 10 times the number of grid points over the *z*-direction grid. Thus, for an example of a 100×100×20 array, with a 10×10 processor array then each individual domain size is 10×10×20.

Figure 6.3 shows a simulation with a 5×4 array. Each processor on the network is assigned a physical domain within the simulation.

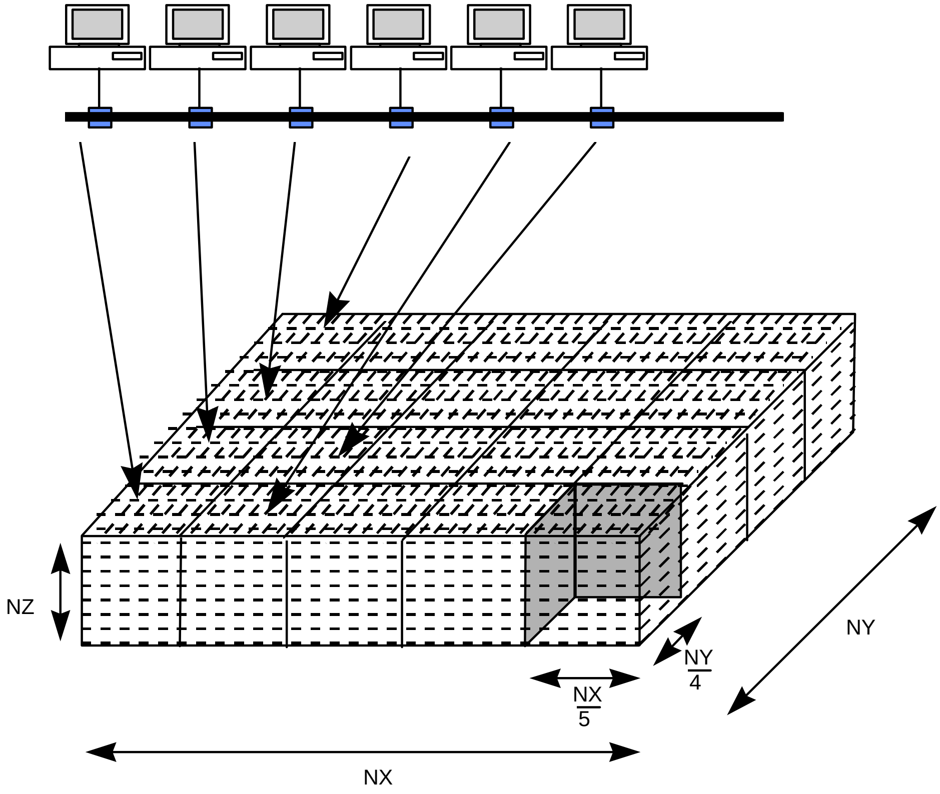


Figure 6.3: Ethernet frame format

In general for a N_x×N_y processor array with NX×NY×NZ grid points, the number of cells transmitted at each interface, in the x-direction, is:

$$NCells_x = \frac{NY}{N_y} NZ \tag{6.1}$$

and the number of cells transmitted at each interface in the *y*-direction is:

$$NCells_y = \frac{NX}{N_x} NZ \tag{6.2}$$

The number of bits transmitted at each interface, in one direction, will then be 32 times these values (assuming 4 bytes per floating point value). As there are six field parameters transmitted for each cell (E_x, E_y, E_z, H_x, H_y and H_z), then the number of frames transmitted at each interface in the *x*-direction ($Nframes_x$) will be:

$$Nframes_x = \text{CEILING}\left(\frac{4 \times 6 \times NCells_x}{1514}\right) \tag{6.3}$$

The number of frames, per interface, in the *y*-direction ($Nframes_y$) is:

$$Nframes_y = \text{CEILING}\left(\frac{4 \times 6 \times NCells_y}{1514}\right) \tag{6.4}$$

The communications overhead for transmission at an interface, in the *x*-direction, is:

$$Overhead_x = Nframes_x \times 122 \times 8 \quad \text{bits} \tag{6.5}$$

The total number of interfaces in the x-direction will be $(N_x-1)\times N_y$ and in the y-direction it is $(N_y-1)\times N_x$. Assuming 4 bytes for each floating point value (32 bits), the total number of bits transmitted for each interface in the x-direction will be:

$$Trans_x = NCells_x \times 32 + Overhead_x \tag{6.6}$$

The number of interfaces transmitted in the x-direction will be:

$$Interfaces_x = 2 \times (N_x - 1) \times N_y \tag{6.7}$$

and the interfaces in the y-direction will be:

$$Interfaces_y = 2 \times (N_y - 1) \times N_x \tag{6.8}$$

Thus the total number of bits transmitted for each iteration will be:

$$Total = Interfaces_x \times Trans_x + Interfaces_y \times Trans_Y \tag{6.9}$$

Substituting equations (6.1) – (6.8) into (6.9) gives the total bits transmitted per iteration:

$$Total = \left[2\times\left(N_x - 1\right)\times N_y\right]\cdot\left[\left(\frac{NY\times NZ\times 32}{N_y}\right) + CEILING\left(\frac{NY\times NZ\times 24}{N_y\times 1514}\right)\times 122\right] +$$
$$\left[2\times\left(N_y - 1\right)\times N_x\right]\cdot\left[\left(\frac{NX\times NZ\times 32}{N_x}\right) + CEILING\left(\frac{NX\times NZ\times 24}{N_x\times 1514}\right)\times 122\right] \text{ bits} \tag{6.10}$$

Since Ethernet uses a 10 Mbps transmission rate, then the time taken to transmit all the boundary conditions will be the total number of bits transmitted per iteration (as given in (6.10)) divided by 10×10^6, that is:

$$T_{comms} = \frac{\textit{Total number of bits transmitted}}{10\times 10^6} \quad \text{s} \tag{6.11}$$

6.4.1 Synchronisation

As all computers on an Ethernet network share the same communications channel then only one of them can transmit at a time. Thus, some form of synchronisation is required so that two or more computers do not talk at the same time. As with the round-robin method discussed in Chapter 5, the computers on the network simulate the problem as if they were connected in an N_x by N_y array. Figure 6.4 shows an example of a 3×3 array, in this case following communications occur:

- computer 1 communicates with computer 2, and vice-versa;
- computer 1 communicates with computer 4, and vice-versa;
- computer 2 communicates with computer 3, and vice-versa;
- computer 2 communicates with computer 5, and vice-versa;
- computer 3 communicates with computer 6, and so on.

6.4.2 Simulation time

Table 6.1 summarises the results for a 100×100×20 grid for 5 000 time iterations using HP 700 workstations connected over a 10 Mbps Ethernet network and Figure 6.5 shows a plot of the total time taken. It can be seen that the simulation time significantly reduces with an increase in array size until the array is larger than 3×3. Above this size the simulation time actually increases, although the total processing time continues to reduce.

The actual total simulation time depends on the processing power of the computers used. The slower the computer, the larger the array size can become before the communication overhead has a great effect.

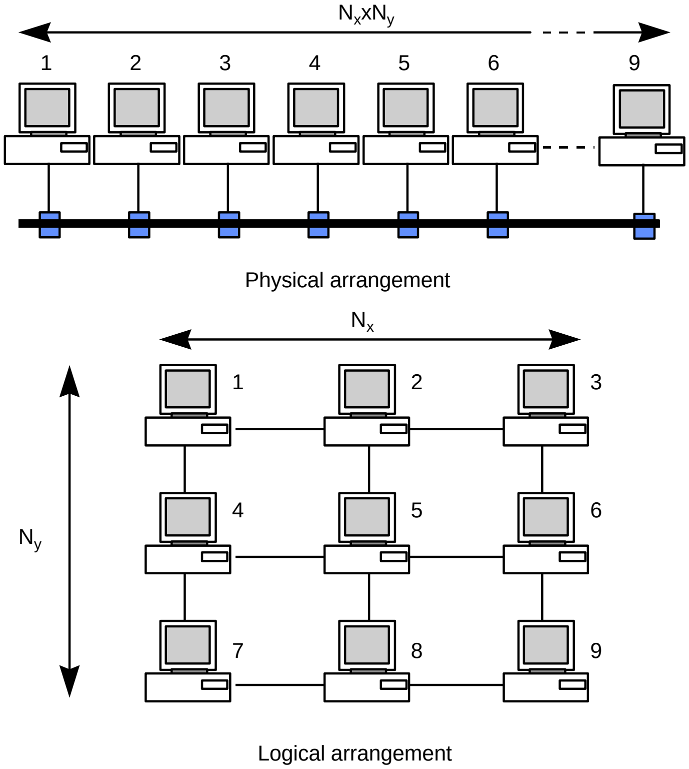


Figure 6.4: Physical and logical arrangement of simulation computers

Table 6.1: Computation time for a 100×100×20 array with 5 000 iterations at 10 Mbps

Processor array	Total communications time (s)	Total processing time (s)	Total simulation time (s)	Transmission overhead (%)
1×1	0	5 000	5 000	0
2×2	136	1 250	1 386	11
3×3	272	556	828	49
4×4	407	313	720	130
5×5	546	200	746	273
6×6	684	139	823	492
7×7	819	102	921	803
8×8	951	78	1 029	1 219
9×9	1 094	62	1 156	1 765
10×10	1 240	50	1 290	2 480

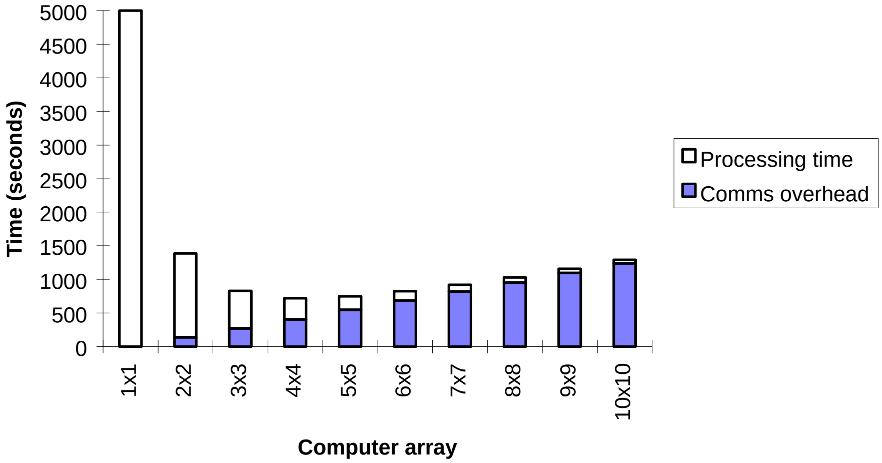


Figure 6.5: Computation time for a 100×100×20 array with 5 000 iterations at 10 Mbps

6.5 Conclusions

The data in Table 6.1 and Figure 6.5 show that for the computer network used and the 3D FDTD method chosen, parallel processing makes a significant effect on the simulation time. The optimum size, in this case, is a 3×3 array.

As discussed in the chapter, the actual total simulation time depends on the processing power of the computers used and the bit rate of the communications channel. The slower the computer, the larger the array size can become before the communication overhead has a great effect.

New ‘fast-Ethernet’ networks, which operate at 100 Mbps, or Fibre Distributed Data Interchange (FDDI) networks, which give an effective bit rate of 200 Mbps, will allow super-fast simulations with a large processor array because the communications overhead reduces by a factor of 10, or 20, for the same problem size.

Special purpose computers could be built for 3D FDTD simulations, but parallel processing over a LAN has the great advantage that the networked computers can be used for other purposes when not simulating the method.

The parallel processing of the 3D FDTD method over networks can be applied to produce extremely large arrays with 10^6 or 10^7 cells, as in [6.1]. These large simulation domains allow large arrays to be built with a relatively low communiations overhead because the processor time per element also remains relatively high.

6.6 References

[6.1] PS Excell, AD Tinniswood, “A Finite-Difference Time-Domain Program for Parallel Computers”, *1995 Antenna Symposium*, Queen Mary & Westfield College, July 1995.

[6.2] Buchanan WJ, “Applied Data and Computer Communications”, *Chapman & Hall*, April 1996.

CHAPTER 7

Results: Propagation in and outside a Microstrip Antenna

7.1 Introduction

Chapters 5 and 6 discussed the parallel processing of the 3D FDTD method, this chapter discusses the application of the method to the simulation of a microstrip antenna. These processing methods were applied in the simulations to reduce run-times. The transient analysis of the antenna is complex, as it involves multiple reflections and is highly resonant.

Chapter 4 discussed the theory of microstrip antennas, whereas Chapter 3 described the application of the 3D FDTD method to microstrip antennas. Balanis and Panayiotis [7.1] applied the 3D FDTD method to model and predict the radiation patterns of wire and aperture structures. Sheen, *et. al.* [7.2] showed how the 3D FDTD method applies to the simulation of a microstrip antenna and Buchanan, Gupta and Arnold [7.3]–[7.8] describe the application of the 3D

FDTD method to the simulation of microstrip antennas.

7.2 Results

This section discusses the results of a simulation using a 4-transputer array connected to a 486-based PC. The simulated antenna has a width of 12.45 mm, a length of 16.00 mm and a feed width of 2.46 mm. This feed is offset from the edge of the antenna head by 2.09 mm. The applied grid is 100×100×12 which gives a time-step of approximately 1.25 picoseconds.

Figures 7.1 – 7.16 show the results for step in time. Figures 7.1 – 7.6 show the electric fields (E_z) under the antenna and Figures 7.7 – 7.16 show the electric field (E_z) just above the antenna.

In Figure 7.1, the gaussian pulse enters the feed to the antenna head. Next, in Figure 7.2, the pulse enters the antenna head and spreads out. A negative pulse is then reflected from the interface between the feed and the antenna head (the antenna head has a lower impedance than the feed). In Figure 7.3, the transmitted pulse continues to spread out in the antenna head and the reflected pulse can be seen to propagate back towards the source. In Figure 7.4, the reflected pulse is absorbed by the source (which is matched to the feed) and the propagating pulse in the antenna head reaches the edges of the antenna. Figure 7.5 shows how the propagating pulse in the antenna head is reflected from the edges (open circuit condition). Not all this energy is reflected, some radiates out from the edges into free-space. Figure 7.6 shows that the electric fields after 1000 ps have decayed to almost zero as the resonance has died away.

Figures 7.7 – 7.16 show the electric fields (E_z) just above the antenna. Figure 7.7 shows the pulse propagating along the feed. Figures 7.8 – 7.15 show the pulse propagating back and forward under the antenna and the leakage of the fields occurring at the edges of the antenna. These figures show the electric fields radiating outwards from the antenna. Figure 7.16 shows the electric fields after 800 ps, by this time most of the energy has been either absorbed by the source or radiated into free space.

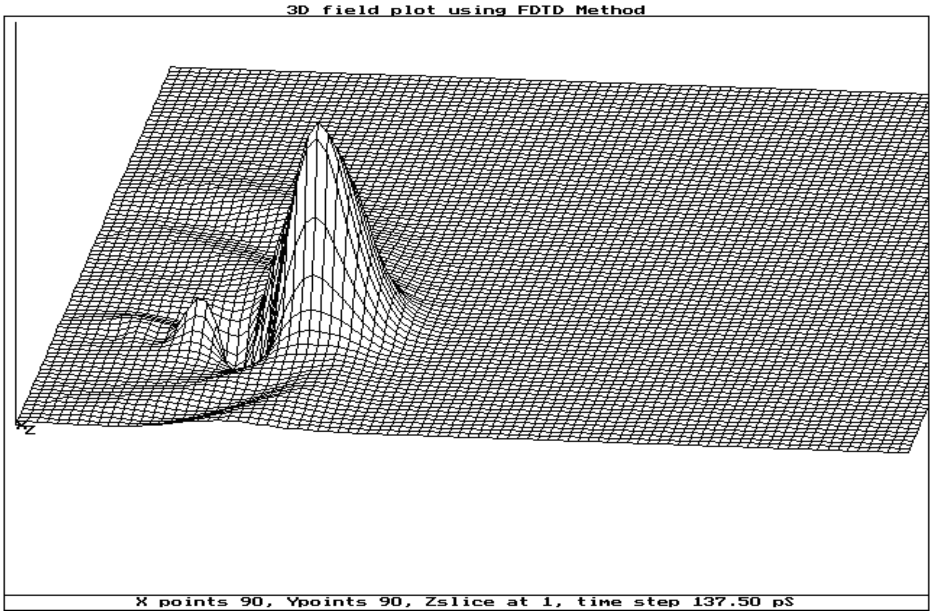


Figure 7.1: Electric field under the antenna at 137.5 ps

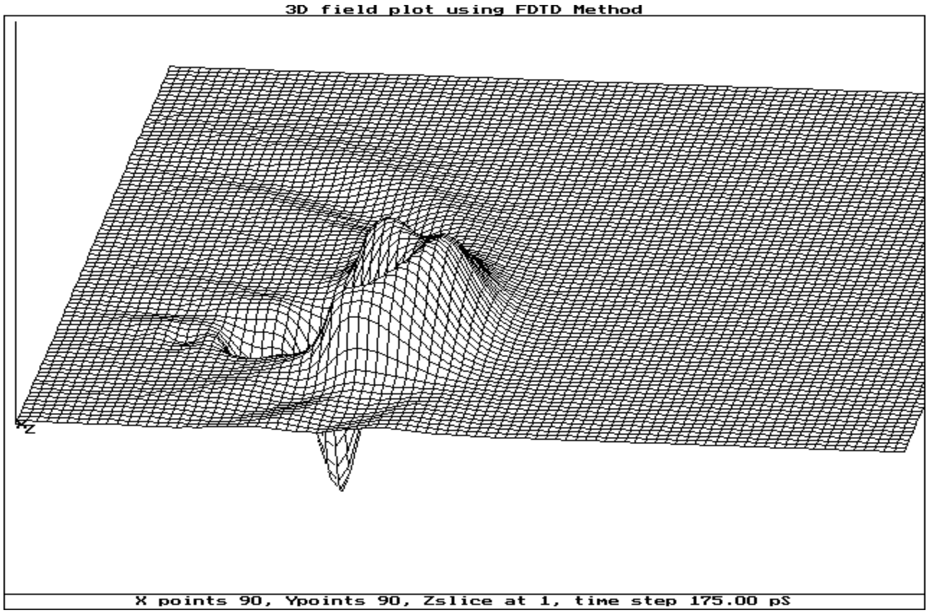


Figure 7.2: Electric field under the antenna 175 ps

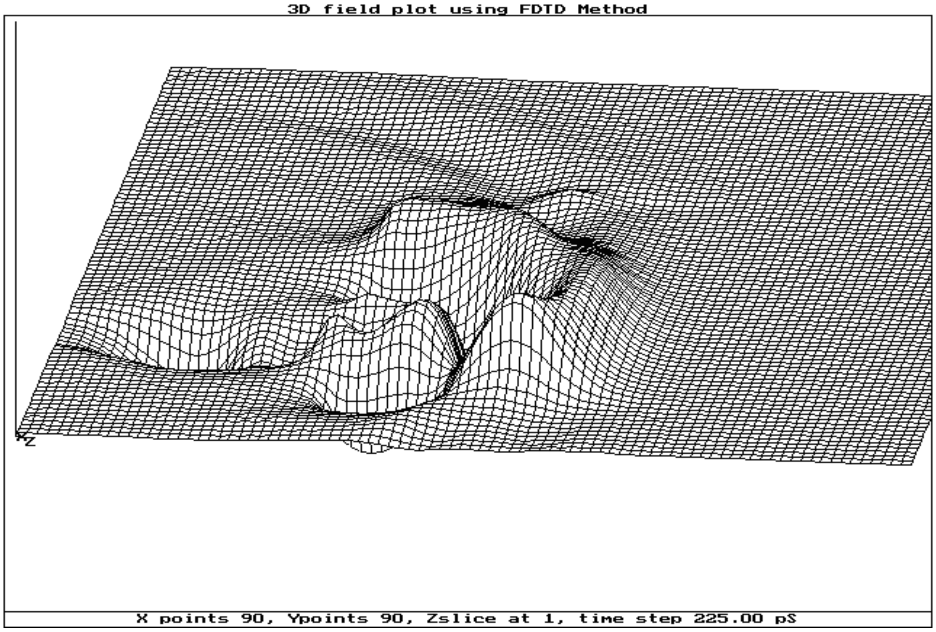


Figure 7.3: Electric field under the antenna at 225 ps

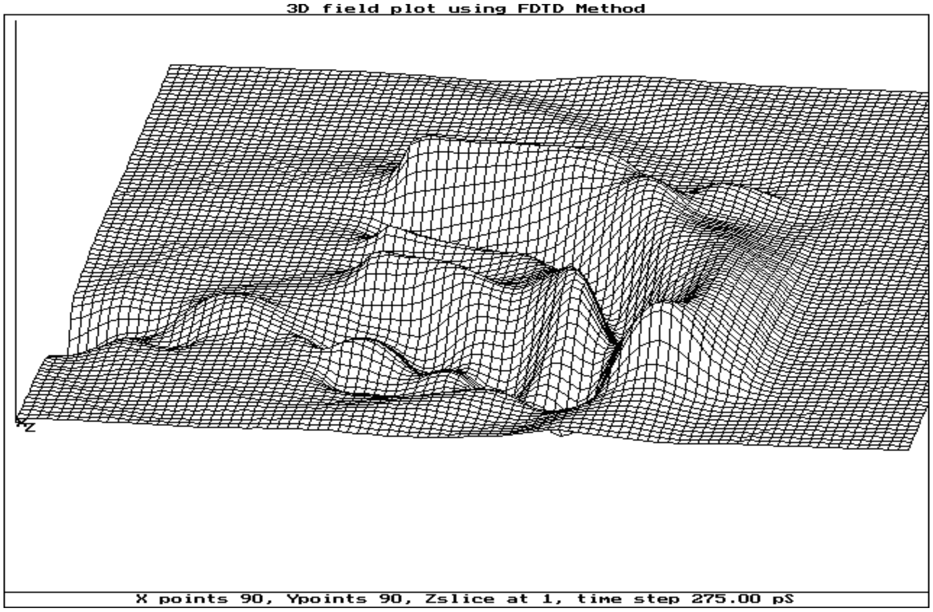


Figure 7.4: Electric field under the antenna at 275 ps

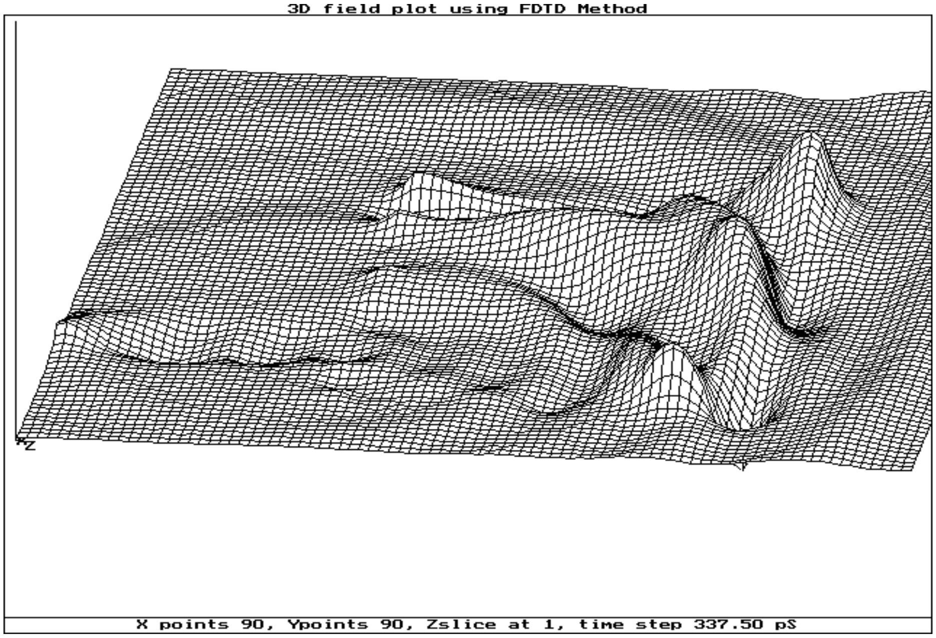


Figure 7.5: Electric field under the antenna at 337 ps

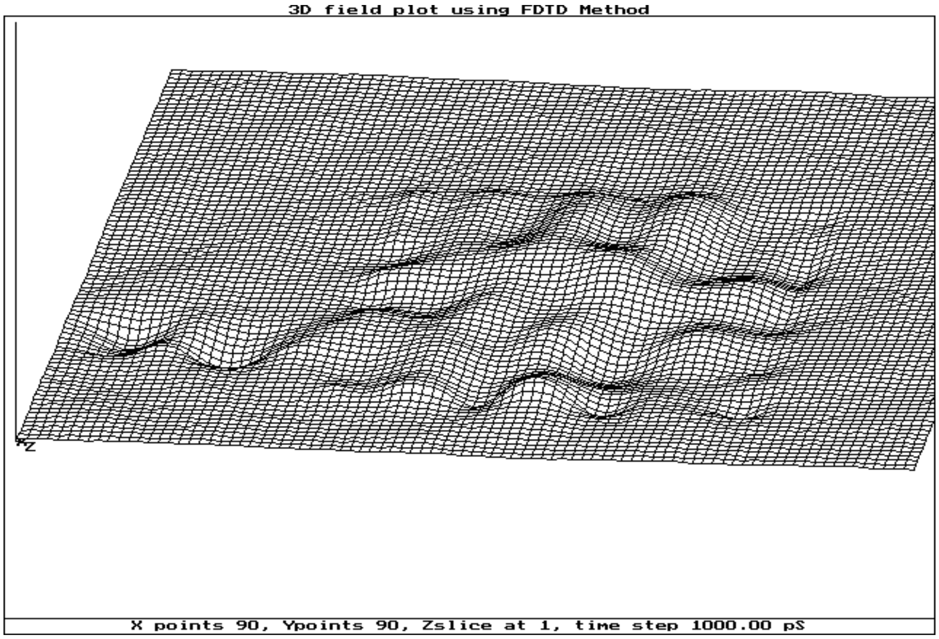


Figure 7.6: Electric field under the antenna at 1000 ps

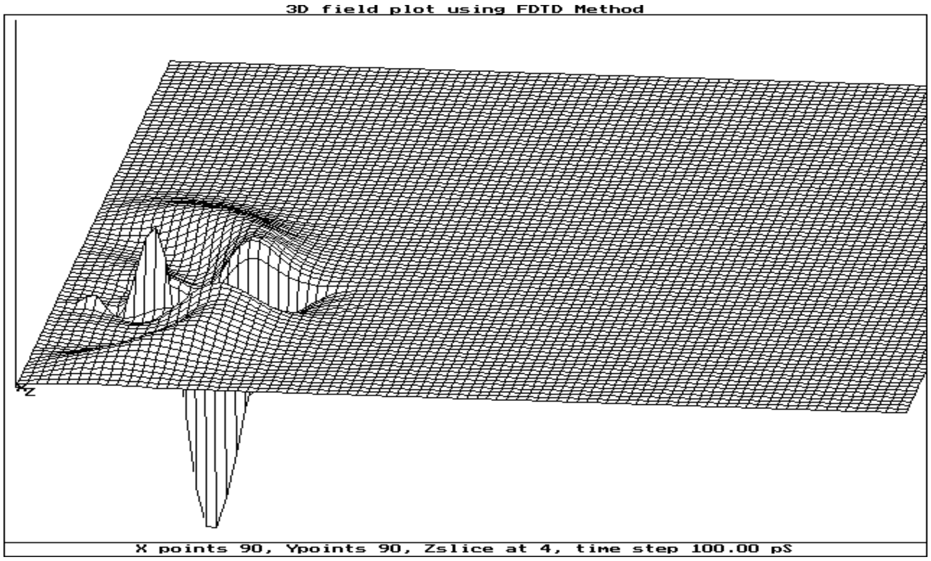


Figure 7.7: Electric field above the antenna at 100 ps

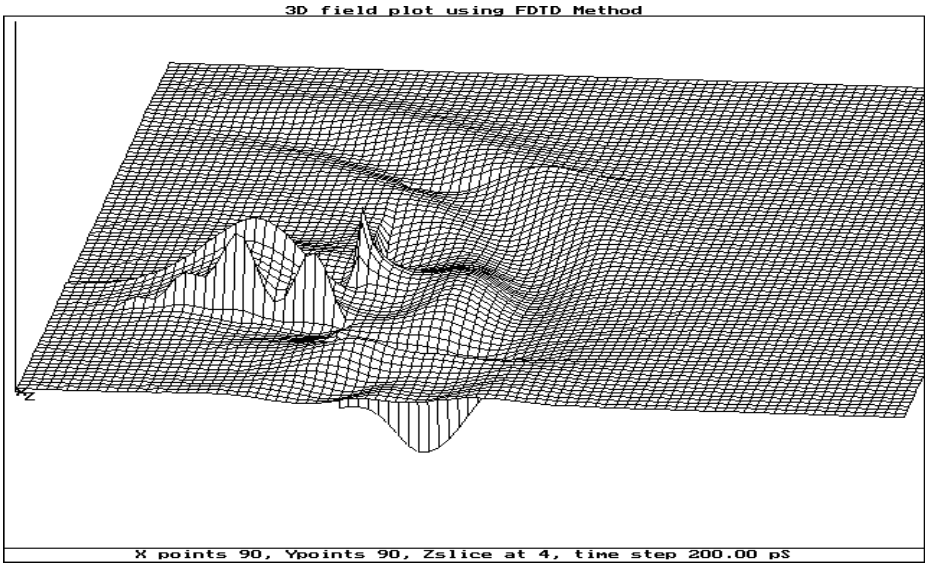


Figure 7.8: Electric field above the antenna at 200 ps

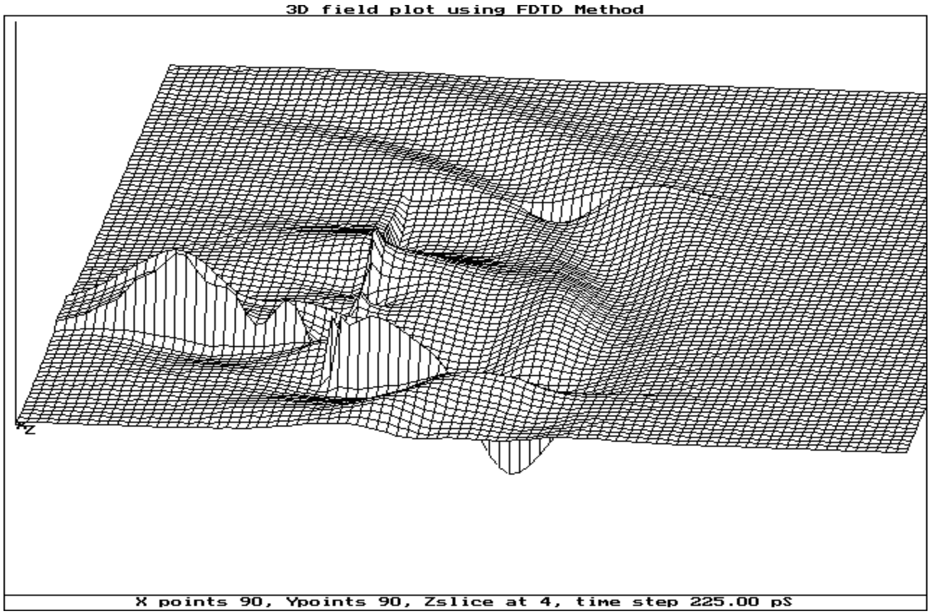


Figure 7.9: Electric field above the antenna at 225 ps

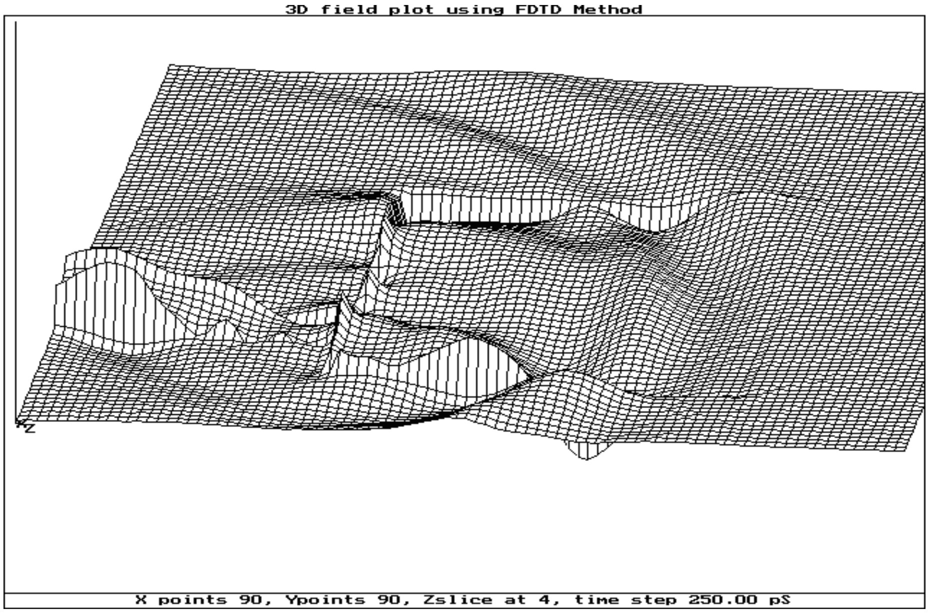


Figure 7.10: Electric field above the antenna at 250 ps

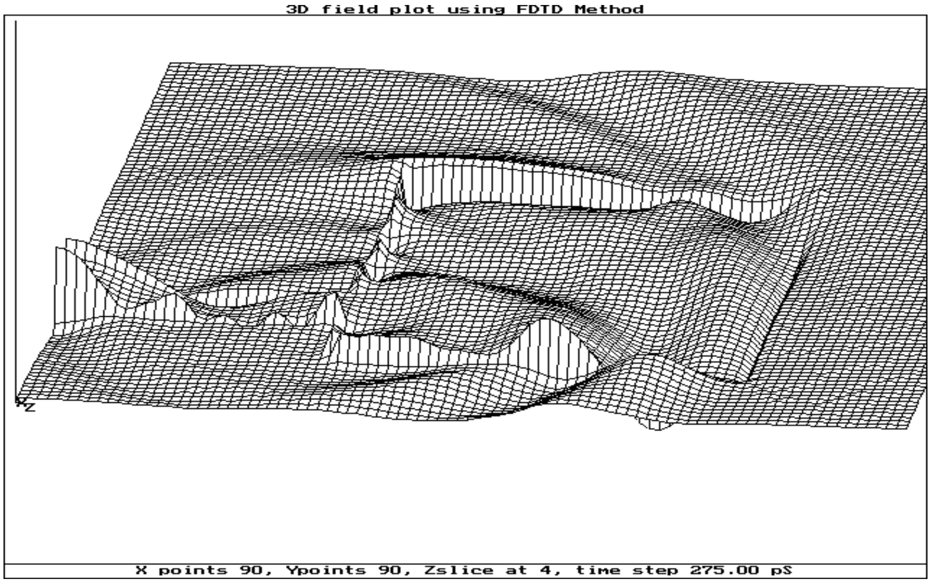


Figure 7.11: Electric field above the antenna at 275 ps

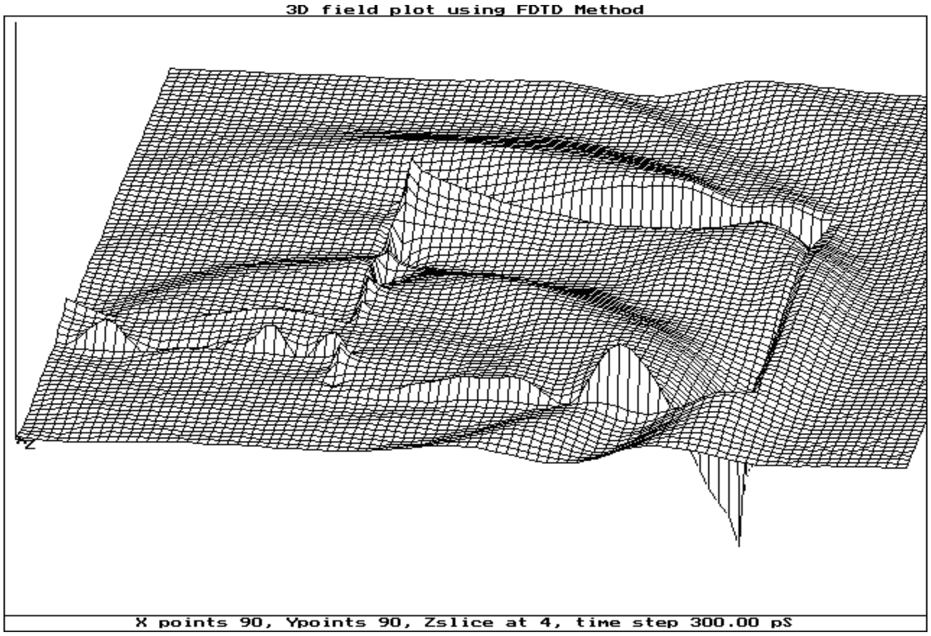


Figure 7.12: Electric field above the antenna at 300 ps

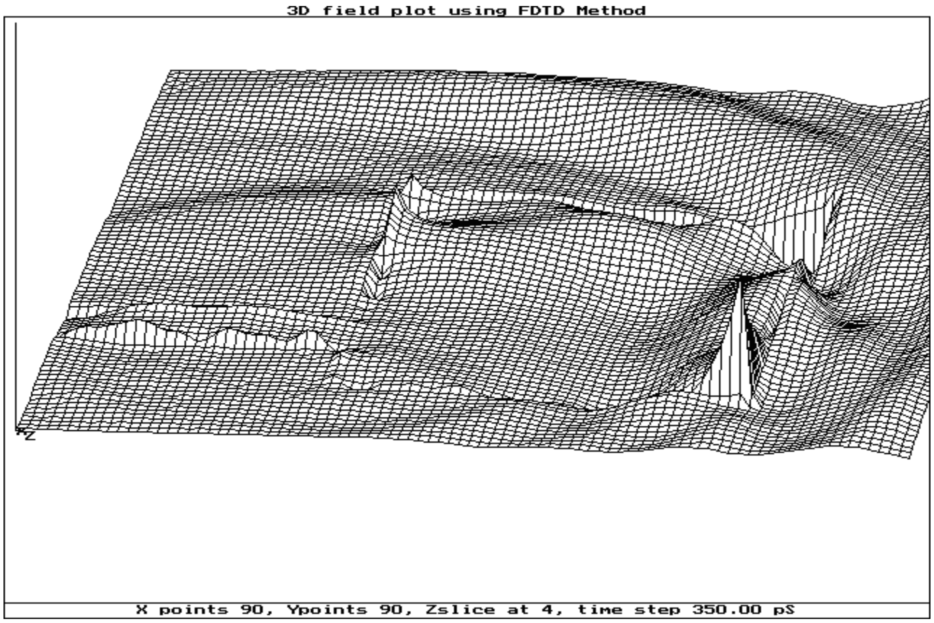


Figure 7.13: Electric field above the antenna at 350 ps

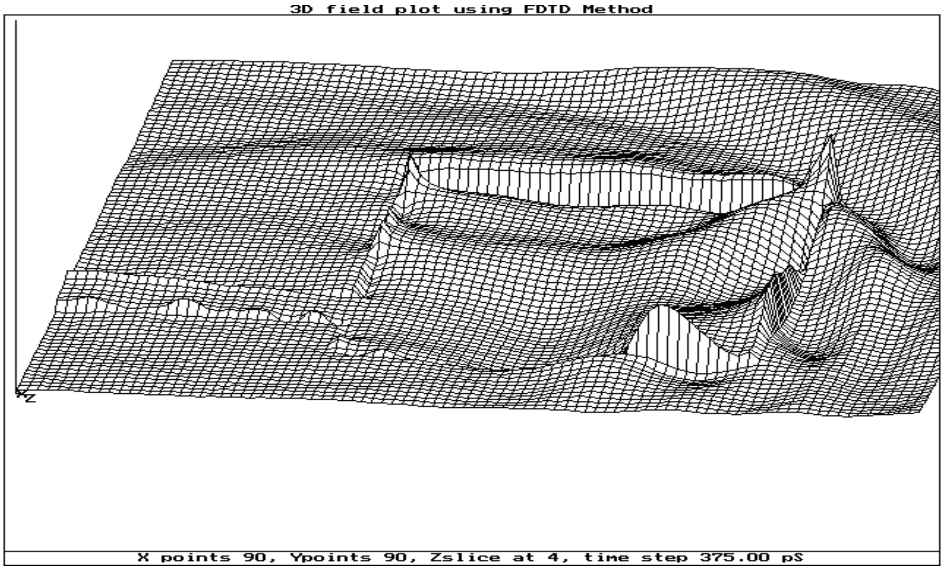


Figure 7.14: Electric field above the antenna at 375 ps

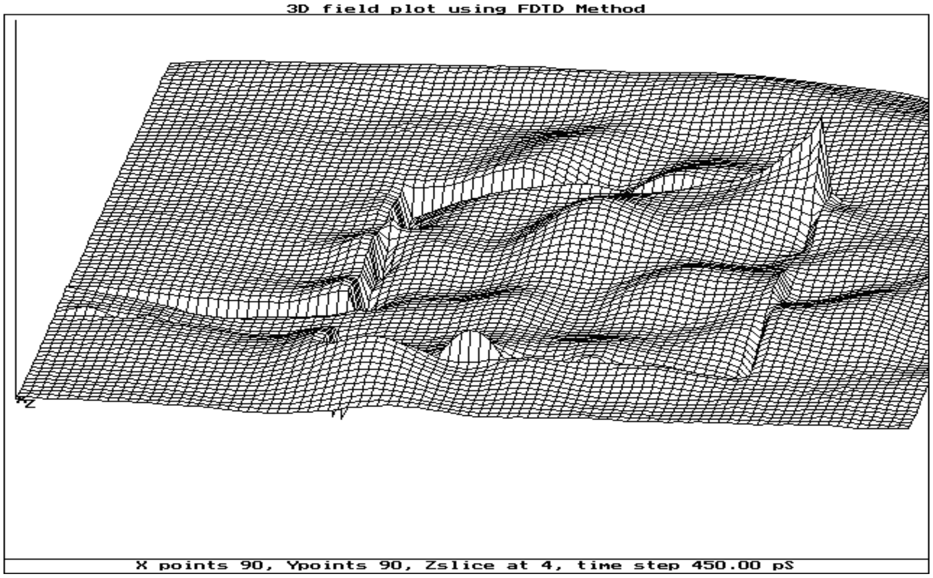


Figure 7.15: Electric field above the antenna at 450 ps

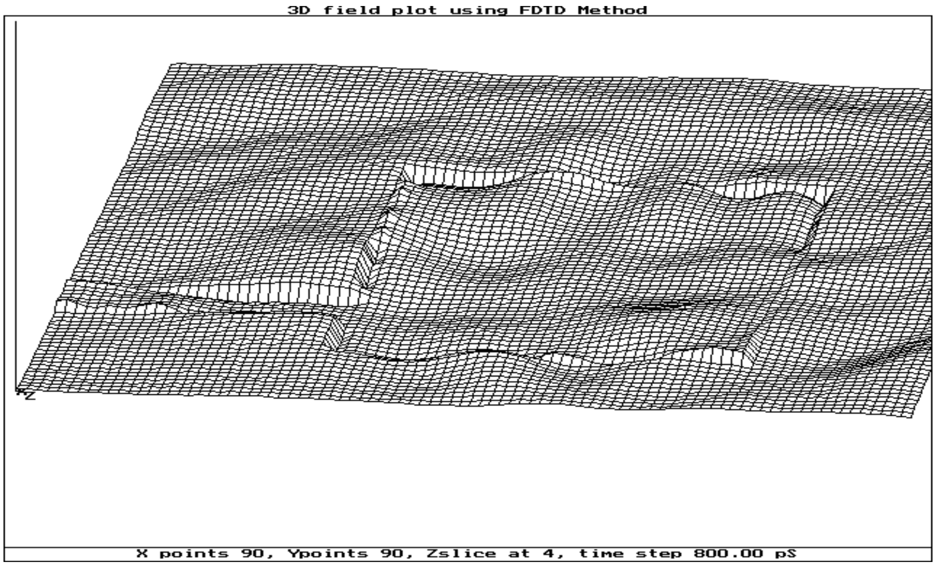


Figure 7.16: Electric field above the antenna at 800 ps

7.3 Analysis of results

The electric field plots in figures 7.1 – 7.6 show the field intensity in the *z*-direction and are measured just below the antenna. The return loss (s_{11}) in Figure 7.17 is a measure of the reflected energy at a given frequency; the less the energy returned the higher the resonated or radiated energy. This assumes that no energy is dissipated within the antenna. A return loss of 0 dB means that all the energy is returned to the source; at –40 dB very little of the incident energy is returned. Figure 7.17 shows that the antenna resonates at, as predicted, 7.5 GHz, as expected and that over 90% of the incident energy is radiated at and around frequencies of 7.5 GHz, 10 GHz, 12 GHz and 18 GHz.

Figure 7.18 shows the radiation pattern for the simulated (dashed line) and expected (solid line) results. The radiation is monitored at a constant radius around the antenna. There is no need for a near field to far field conversion as if the fields are monitored at points where the near field have little effect. In this case of this antenna the fields are monitored at a contant radius of 20 mm. It can be seen from this that the radiation pattern for the antenna is a good match with the expected results, using [4.11] and [4.12].

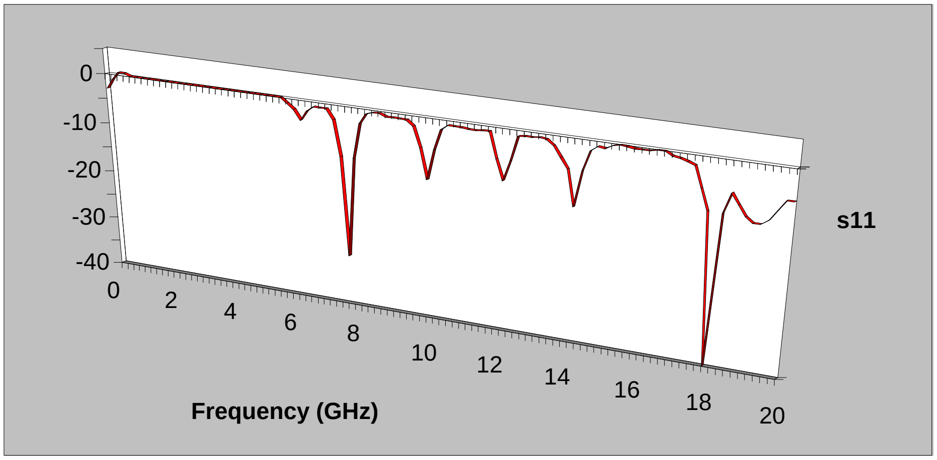


Figure 7.17: Return loss (in dBs) from antenna

7.4 Conclusion

The simulation in this chapter shows the propagation of a pulse within a microstrip antenna, and the electric fields under and just above the antenna. It has been seen that the 3D FDTD method is a good technique for predicting electric field propagation. The technique can be used to generate wide frequency responses with no change in modelling. It also provides a near complete solution of Maxwell’s equations in a 3D model.

As computers become faster and memory storage greater, larger models can be simulated with greater accuracy as compared to other empirical methods.

One disadvantage of the 3D FDTD method is that it fails to take into account losses in the dielectric and non-perfect conductors.

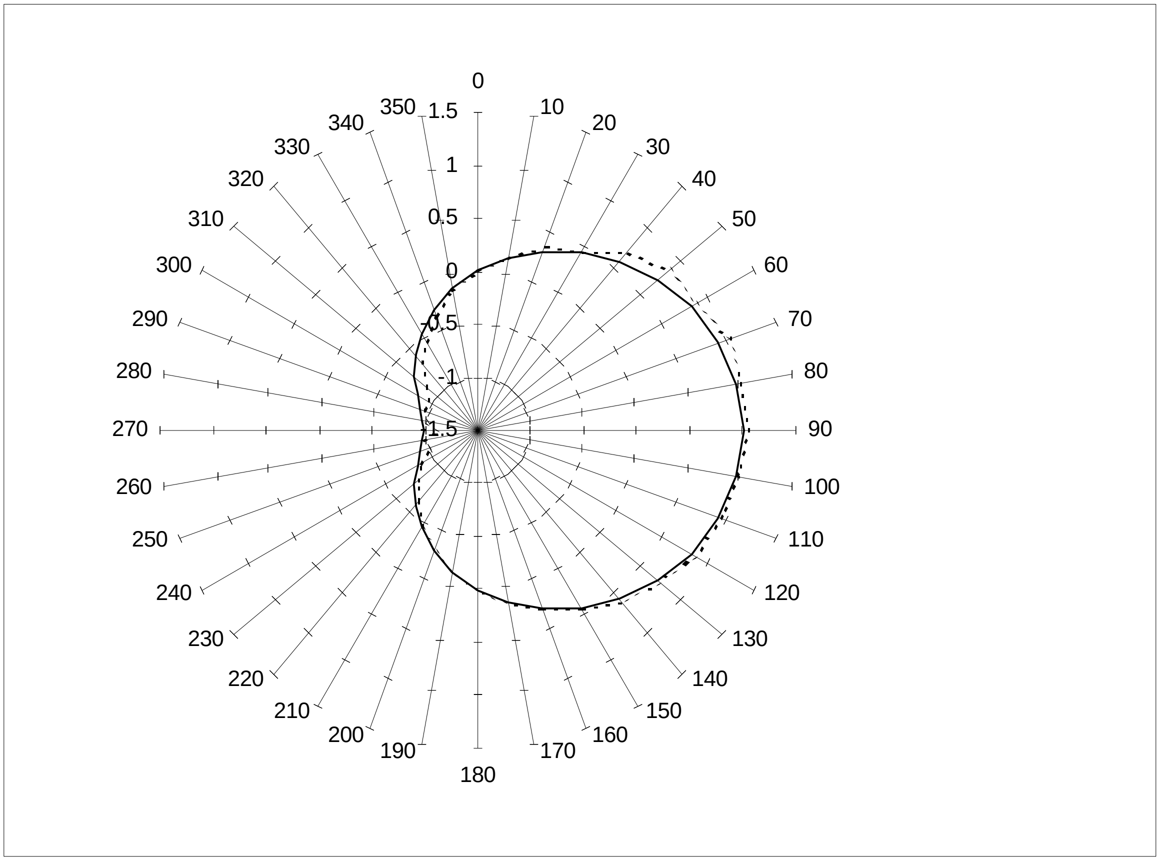


Figure 7.18: Radiation pattern from the antenna

7.5 References

[7.1] Tirkas PA and Balanis CA, "Finite-Difference Time-Domain Method for Antenna Radiation", *IEEE Trans. on Antennas and Propagation*, vol. 40, 3, pp 334–857, March 1992.

[7.2] Sheen D, Ali S, Abouzahra M, and Kong J, "Application of Three-Dimensional Finite-Difference Method to the Analysis of Planar Microstrip Circuits", *IEEE MTT*, vol. 38, 7, pp. 849–857, July 1990.

[7.3] Buchanan WJ, Gupta NK, "Simulation of Near-Field Radiation for a Microstrip Antenna using the 3D-FDTD Method", *NRSC '93*, University of Leeds, Apr. 1993.

[7.4] Buchanan WJ, Gupta NK and Arnold JM, "Simulation of Radiation from a Microstrip Antenna using Three-Dimensional Finite-Difference Time-Domain (FDTD) Method", *IEE Eight International Conference on Antennas and Propagation*, Heriot-Watt University, Apr.

1993.

[7.5] Buchanan WJ, Gupta NK and Arnold JM, "3D FDTD Method in a Microstrip Antenna's Near-Field Simulation", *Second International Conference on Computation in Electromagnetics*, Apr. 1994.

[7.6] Buchanan WJ, Gupta NK and Arnold JM, "Application of 3D Finite-Difference Time-Domain (FDTD) Method to Predict Radiation from a PCB with High Speed Pulse Propagation", *Ninth International Conference on Electromagnetic Compatibility*, University of Manchester, UK, Aug. 1994.

[7.7] Buchanan WJ, Gupta NK, "An Accurate Model for the Parallel Processing of the 3D Finite-Difference Time Domain (FDTD) Method in the Simulation of Antenna Radiation", *QMW 1996 Antenna Symposium*, Jul. 1995.

CHAPTER 8

Results: EM Fields in a PCB

8.1 Introduction

This chapter discusses the simulation of the propagation of electromagnetic waves within and outside a printed circuit board (PCB). This information is important in the design of electronic systems as they must now comply with EC Electromagnetic Compatibility (EMC) regulations. Most current methods involve building prototypes of systems and testing them to determine if they meet the EMC regulations. A better solution is to simulate the system by computer and modify the design so that it complies with the regulations. It is the intention of this chapter to discuss the application of the 3D FDTD method to the simulation of EMC from a PCB.

Typical simulation methods used in the simulation of EMC are the Method of Moments, the Transmission Line Method [8.1], and Finite Element methods [8.2]. Unfortunately, these methods are not directly formulated from Maxwell's equations, thus the 3D FDTD method has great potential.

The 3D FDTD method produces the transient analysis of the PCB which takes into account reflections from mismatches on the tracks and inter-coupling of the electrical signals. Chapter 3 discussed the application of the 3D FDTD method to microstrip antennas. These methods can be easily adapted to the simulation of PCBs. The parallel techniques discussed in chapters 6 and 7 were used to reduce simulation times.

Railton CJ and McGeehan JP [8.3], Pothecary N and Railton CJ [8.4] and Buchanan, *et. al.* [8.5] – [8.7] outline the application of the 3D FDTD method to determine the radiation and cross-talk from PCBs.

8.2 Simulated model

The simulated PCB has a width of 38.9 mm, a length of 40 mm and a substrate thickness of 0.8 mm. A 100×100×16 grid was used giving a time-step of approximately 0.5 picoseconds.

Figures 8.3 – 8.12 show the electric field in the *z*-direction. just above and below the copper tracks. The results were obtained using a 4-transputer array connected to a 486-based PC.

As expected the simulation time was reduced to almost one-quarter of that for the equivalent single processor. A relatively small amount of time was thus spent with inter-transputer communications.

8.3 Results

Figure 8.1 shows the track structure of the model and the four Gaussian pulse sources. The field plot in Figure 8.3 shows that the pulses have entered the structure and are propagating along the tracks A, C, D and G. In Figure 8.4, the pulses within tracks C and D encounter track E. Next, in Figure 8.5, two negative pulses are reflected from track E and propagate back along tracks C and D.

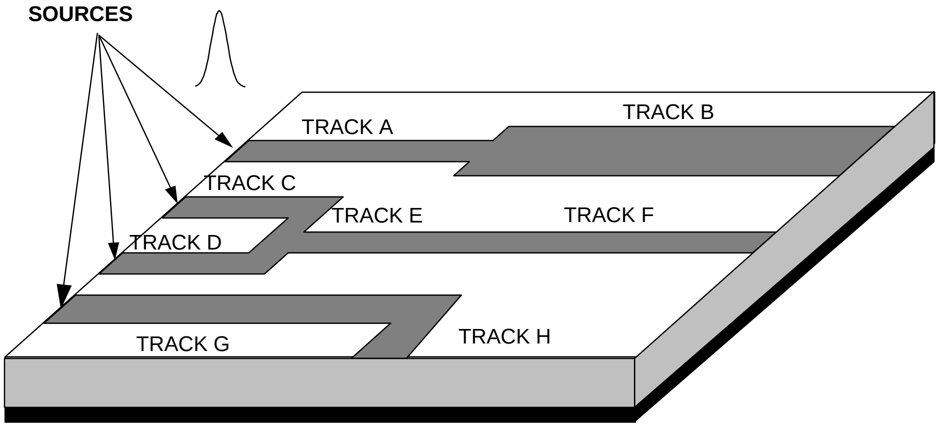


Figure 8.1: Conductor treatment

In Figure 8.6 it can be seen that the pulse in track A enters track B and spreads outwards. A negative pulse is then reflected back from the junction between A and B (the impedance of track B is less than that of A). Figure 8.7 then shows that the pulse travelling in track G changes direction and travels along track H. Figure 8.8 shows a negative pulse travelling back along track A and the pulse travelling in track G being absorbed at the near-side wall. Figure 8.9 shows that after 820 time-steps all the energy in the model has been absorbed.

Figures 8.10 – 8.12 show the electric field just above the PCB. The *z*-component of the electric field directly above the tracks will be negative as the lines of electric field point into the conductors, as illustrated in Figure 8.2.

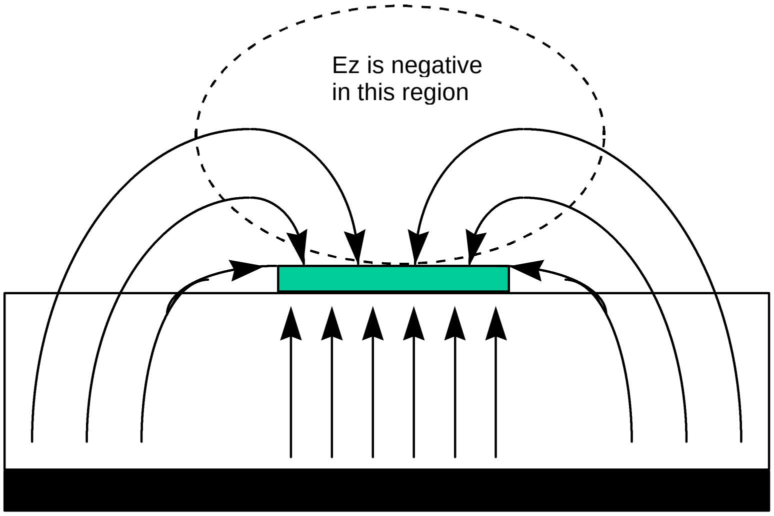


Figure 8.2: Electric fields around a track

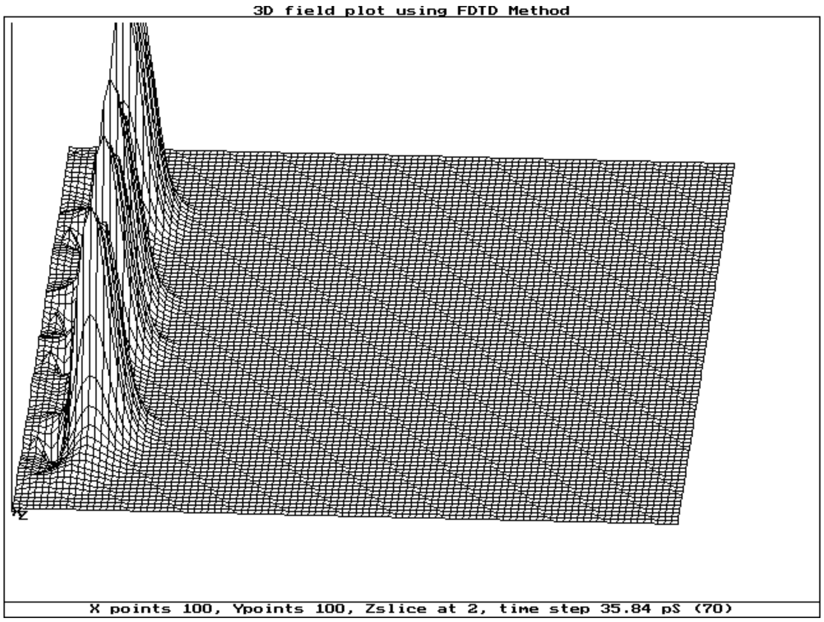
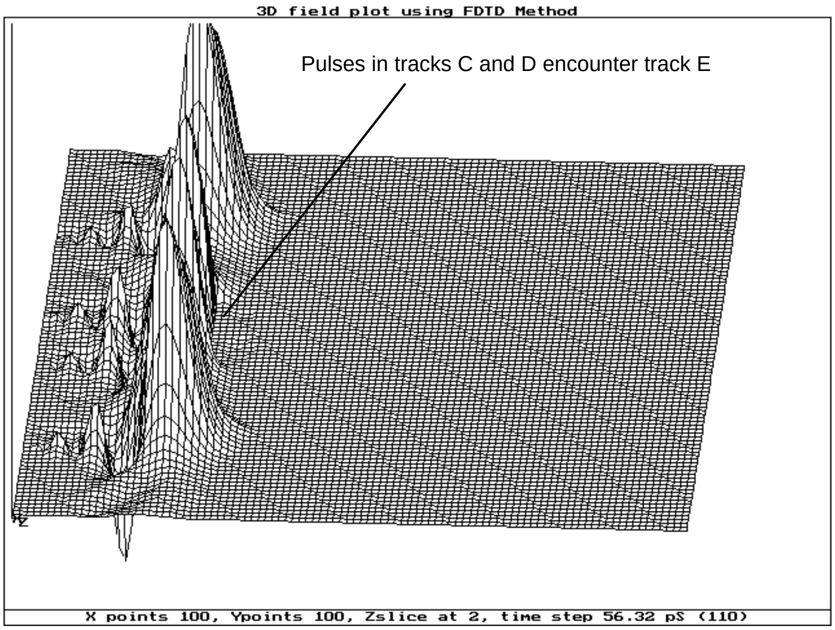
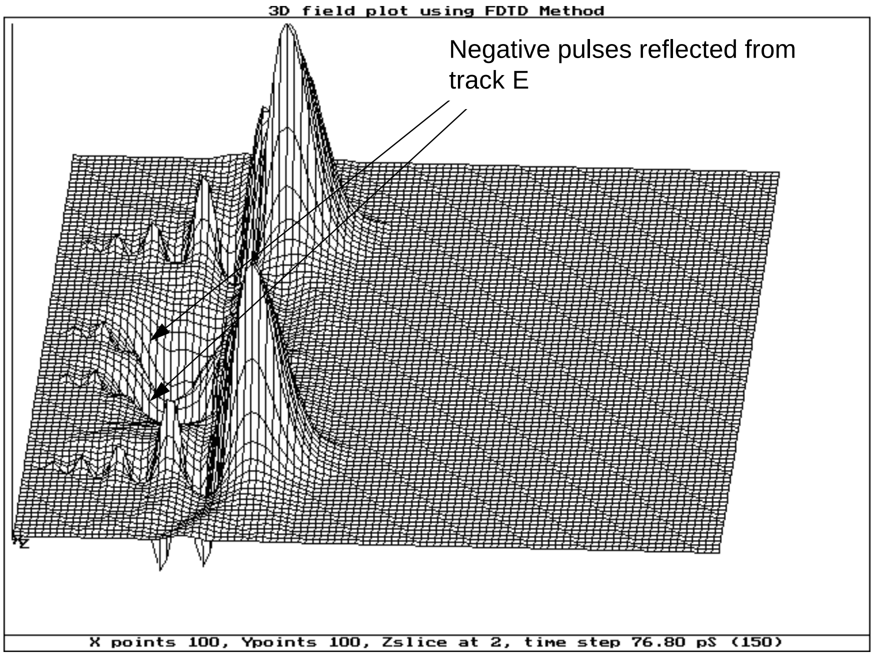


Figure 8.3: E-field within substrate after 70 time steps



Pulses in tracks C and D encounter track E

Figure 8.4: E-field within substrate after 110 time steps



Negative pulses reflected from track E

Figure 8.5: E-field within substrate after 150 time steps

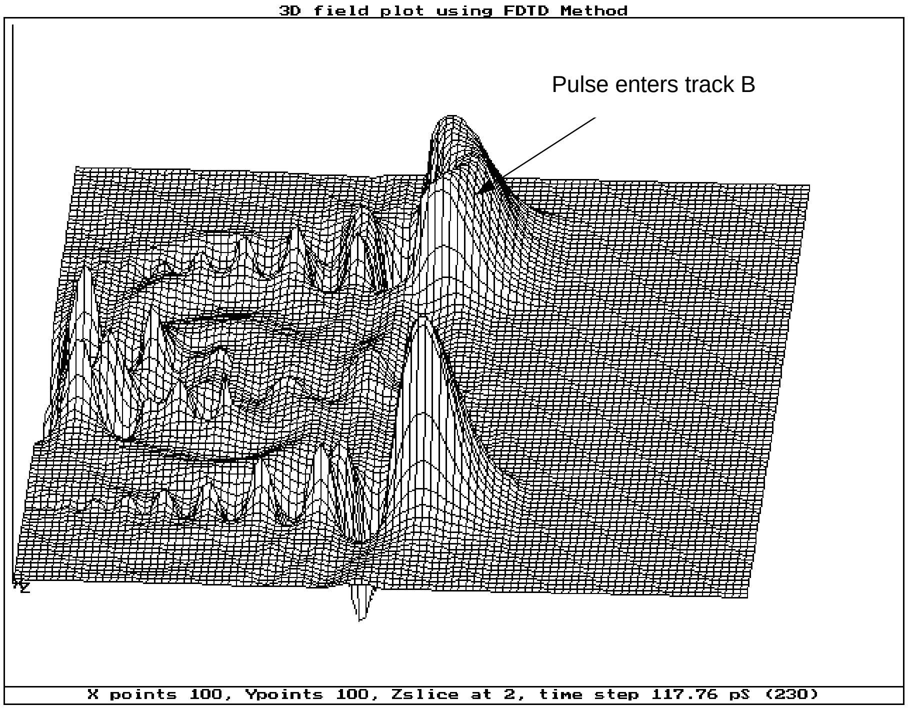


Figure 8.6: E-field within substrate after 230 time steps

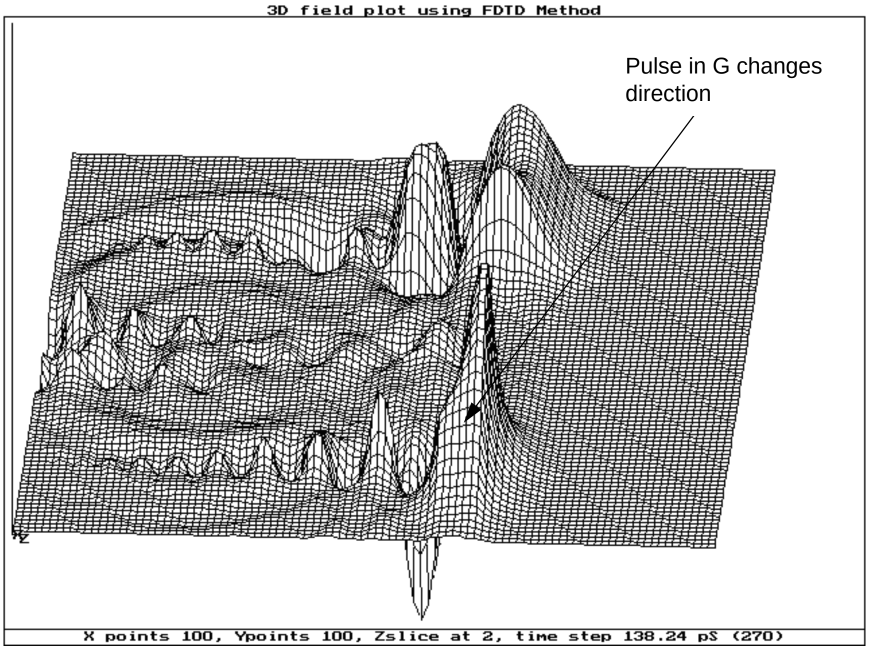
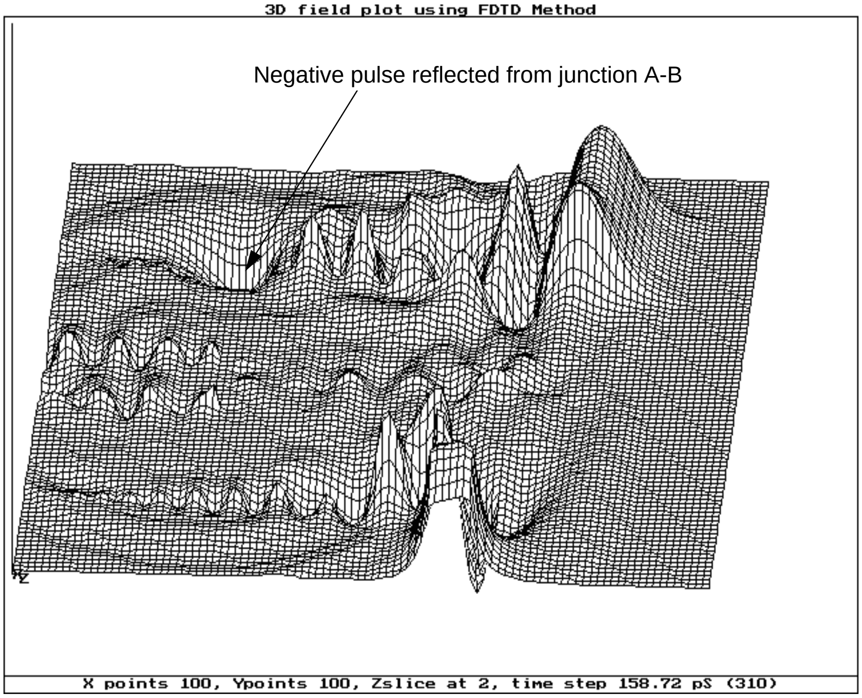


Figure 8.7: E-field within substrate after 270 time steps



Negative pulse reflected from junction A-B

Figure 8.8: E-field within substrate after 310 time steps

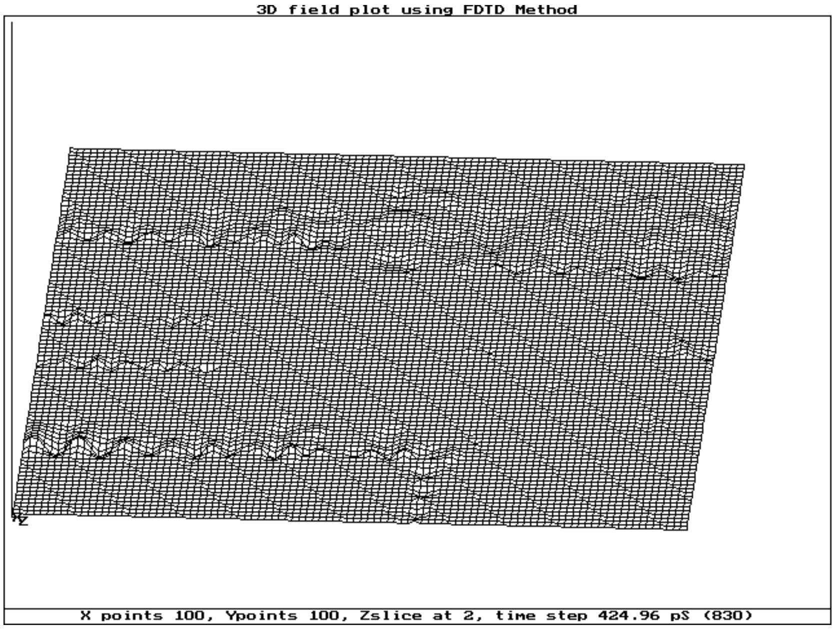


Figure 8.9: E-field within substrate after 820 time steps

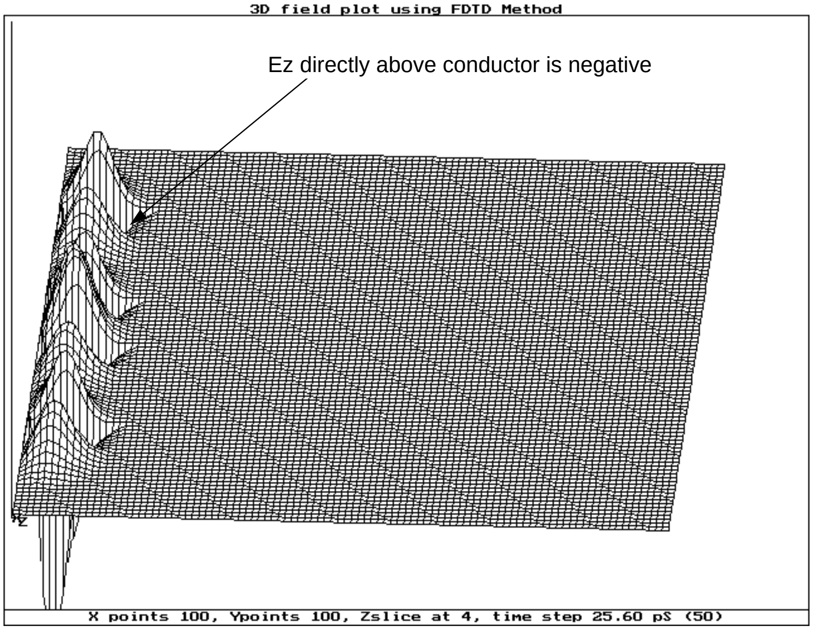


Figure 8.10: E-field above PCB after 50 time steps

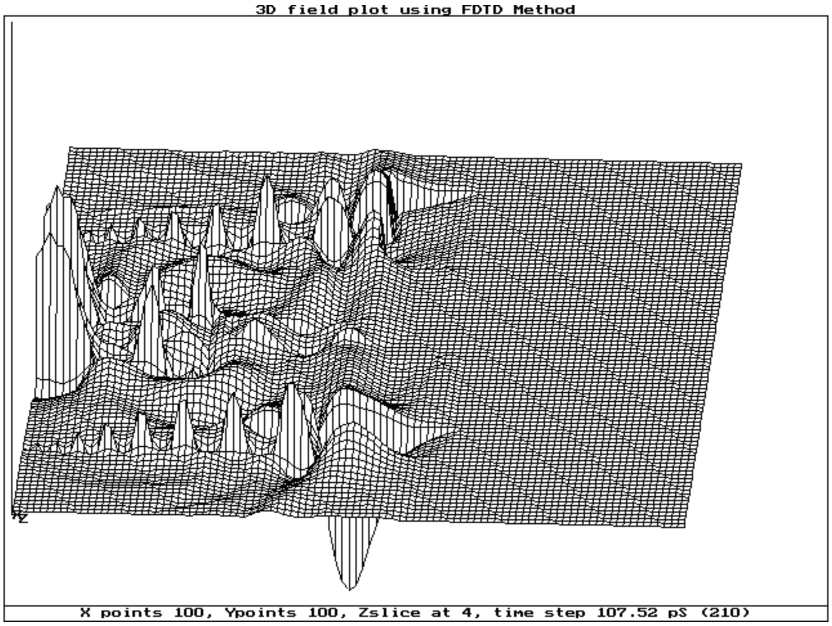


Figure 8.11: E-field above PCB after 210 time steps

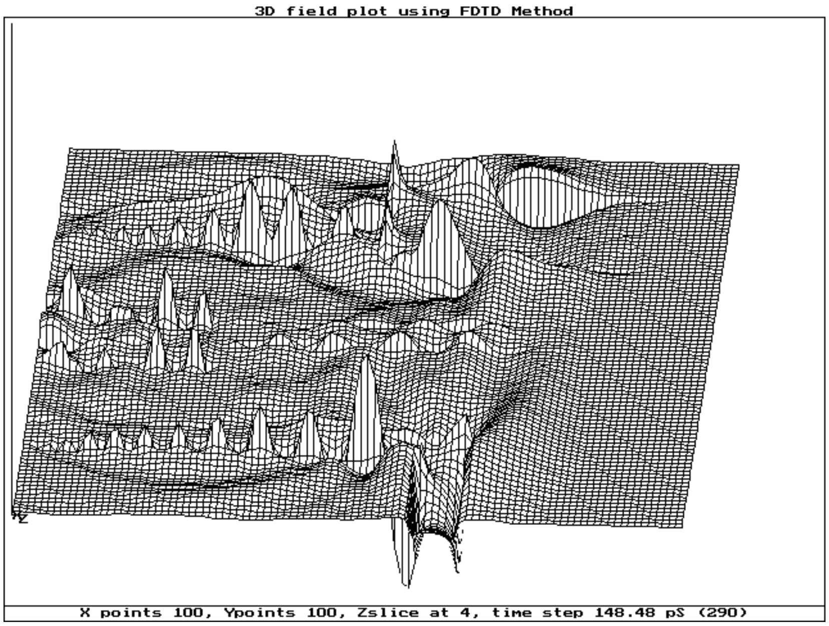


Figure 8.12: E-field above PCB after 230 time steps

8.4 Conclusions

A novel application of the FDTD method has been shown in simulating the propagation of Gaussian pulses applied from multiple sources within and outside a PCB. This has proved useful in showing that the electric field directly above a conductor is negative (that is, pointing towards the conductor).

A disadvantage of the FDTD method is that it simulates structures in the time-domain. This requires a large memory storage and large run-times. However, this problem can be reduced by using modern powerful computers and for very large and complex simulations the use of parallel processing further alleviates this problem.

The results obtained clearly show the propagation and reflection of Gaussian pulses appropriate to their position in the structure and time. These can be used to determine the frequency characteristics of the structure, from DC to the required upper frequency with no change of model for different frequency spectra.

The model used assumes a match between the source and the copper tracks and an absorbing boundary around on the outer walls of the problem. These values will not be totally accurate as the FDTD method does not take into account conduction or dielectric losses.

8.5 References

[8.1] Johns PB, “Use of Condensed and Symmetrical TLM Nodes in Computer Aided Electromagnetic Design”, *IEEE MTT*, 6, pp. 368–374.

[8.2] NEC, STRIPES and MSC / EMAS software packages.

[8.3] Pothecary N and Railton CJ, “Rigorous analysis of cross-talk on high speed digital circuits using the Finite Difference Time Domain Method”, *International Journal on Numerical Modelling*, part H, 6, pp. 368–374.

[8.4] Railton CJ, Richardson KM, McGeehan and Elder KF, “The Prediction of Radiation Levels from Printed Circuit Boards by means of the Finite-Difference Time-Domain Method”, *International Conference on Computation in Electromagnetics*, Nov. 1991, pp. 339–341.

[8.5] Buchanan WJ, Gupta NK, “Simulation of Electromagnetic Pulse Propagation in Three-Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques”, *Electrosoft ‘93*, Jul. 1993, Southampton.

[8.6] Buchanan WJ, Gupta NK and Arnold JM, “Application of 3D Finite-Difference Time-Domain (FDTD) Method to Predict Radiation from a PCB with High Speed Pulse Propagation”, *Ninth International Conference on Electromagnetic Compatibility*, University of Manchester, Aug. 1994.

[8.7] Buchanan WJ, Gupta NK, “Prediction of Electric Fields from Conductors on a PCB by 3D Finite-Difference Time-Domain Method”, *IEE Engineering, Science and Education Journal*, Aug. 1995.

CHAPTER 9

Conclusions

9.1 Achievement of Aims and Objectives

The work covered in this thesis shows only a small part of the work achieved over the registration period. Much, undocumented, work went into areas such as the generation of wire-frame models, animation software, automated data file software, the fourier transform software, and so on. This work has not been included in the main body of the thesis because it would spoil the flow of the text.

At the end of any project it is essential to determine if the results match the initial aims. To summarise, the main objectives of the research were to:

- Investigate frequency- and time-domain methods in the simulation of electromagnetic propagation;
- Model the propagation of electrical signals within microstrip antennas and

printed circuit boards (PCBs) using the three-dimensional (3D) Finite-Difference Time-Domain (3D FDTD) method;

- Determine the electrical characteristics of microstrip antennas and PCBs using the 3D FDTD method;
- Investigate the application of parallel processing to simulations using the 3D FDTD method.

Each of these objectives were met and the author feels that the work outlined in the thesis has led to important breakthroughs and motivated other researchers into this interesting area of electromagnetic simulation.

9.2 Discussion

Chapter 2 discussed the main simulation methods used in EM field modelling. It shows that the FDTD and TLM methods are both time-based and involve stepping a model through discrete intervals in time. The main difference between them is that the TLM method models uses equivalent transmission line elements, whereas, the FDTD method models propagation through the elements using a discrete form of Maxwell’s curl equations.

Chapter 2 concluded that, in general, time-domain solutions have the advantages over frequency-domain solution in that that they provide wide bandwidth responses and they can be used in parallel processing with reduced simulation times. For these reasons the 3D FDTD method was chosen as the main simulation method in this research. The increasing usage of the 3D FDTD method shows that this decision was correct.

Chapter 3 discussed the 3D FDTD method and showed how improvements can be made to FDTD which either improve accuracy or reduce simulation time.

These methods included the sub-gridding method around discontinuities and a the incorporation of static field solutions and Z-transforms into the FDTD method. Improvements can also be made to the boundary conditions and the modelling of sources.

The two structures chosen to model where microstrip antennas and PCBs. Microstrip antennas are extremely difficult to simulation with a time-based solution because they are highly resonant structures. Chapter 4 gave some background theory of these antennas.

Chapter 5 showed how transputer arrays can be used to simulate 3D FDTD problems. Novel equations are presented which can be applied to any multi-processor system connected in a grid array. Figures 5.9 - 5.11 show graphs of simulation times to show the effect of inter-processor communications. These show that transputer arrays are efficient in their parallelism but suffer from a significant communication overhead when connected in large arrays. To overcome this a novel synchronisation method is presented in Section 5.5. This method significantly reduces the inter-processor communication times and thus reduces simulation times. The chapter also shows that higher inter-processor link bit rates also significantly reduce transmission overheads (Figure 5.12).

Chapter 6 extended the parallel processing of the 3D FDTD method over general-purpose clustered workstations connected over an Ethernet network. The chapter derives novel equations for communications and simulation times for. These quations can be used on any system that uses a domain array of processors. Figure 6.5 contrasts simulation times for practical multi-workstations connected over by an Ethernet In general, the actual total simulation time depends on the processing power of the computers used and the bit rate of the communications channel. The slower the computer, the larger the array size can become before the communication overhead has a great effect.

In conclusion, the parallel processing of the 3D FDTD method can either be achieved over a LAN, with transputer arrays or with specially designed parallel computers. New ‘fast-Ethernet’ networks, which operate at 100 Mbps, or Fibre Distributed Data Interchange (FDDI) networks, which give an effective bit rate of 200 Mbps, will allow super-fast simulations with a large processor arrays.

Special purpose computers also could be built for 3D FDTD simulations, but parallel processing over a LAN has the great advantage that the networked computers can be used for other purposes when not simulating the method.

The parallel processing of the 3D FDTD method over networks can be applied to produce extremely large arrays with 10^6 or 10^7 cells. These large simulation domains allow large arrays to be built with a relatively low communications overhead because the processor time per element also remains relatively high.

Chapters 7 discussed the simulation of radiation and propagation in a microstrip antenna and within a PCB. Figures 7.1-7.6 show the propagation of a gaussian pulse within the antenna, from this the return loss for the antenna can be determined (Figure 7.17). Also by monitoring the electric fields above the antenna simulations the radiation pattern around the antenna can be determined. These simulations are shown in figures 7.7 - 7.16 and the resultant field pattern is shown in Figure 7.18. The simulated field pattern compares well with the computed value, as given by equations (4.11) and (4.12). An advantage with the 3D FDTD simulation is that there is no need for a near- to far-field conversion if the field points are monitored at points far enough away from the antenna (in this case, only 20 mm away from the antenna).

Chapter 8 discussed the simulation of the propagation of electromagnetic waves within and outside a printed circuit board (PCB). Simulations can be conducted with multiple source placed at any point on the simulated model. It shows that the 3D FDTD method can be used to investigate the propagation of electrical signals. Future work can investigate the cross-talk between signal lines and the radiation from conductors.

A disadvantage of the FDTD method is that it simulates structures in the time-domain. This requires a large memory storage and large run-times. However, this problem can be reduced by using modern powerful computers and for very large and complex simulations the use of parallel processing further alleviates this problem.

The model used assumes a match between the source and the copper tracks and an absorbing boundary around on the outer walls of the problem. These values will not be totally accurate as the FDTD method does not take into account conduction or dielectric losses.

APPENDIX A

3D FDTD package

A.1 Introduction

In the course of this research many ANSI-C computer programs were developed to model, simulate and analyse electromagnetic systems. Each program was self contained and most of them produced data files for other programs. The main programs developed were:

- A microstrip antenna modeller;
- An automatic data file generator for the 3D FDTD modeller;
- A 3D FDTD modeller;
- A FFT analysis program;
- A 3D EM Field visualiser.

Figure A.1 show the programs used at each of the simulation and analysis steps. The main advantage of splitting the simulation into steps, rather that developing a completely integrated package, is that any simulation step can be implemented on any kind of computer with any type of operation system. For example, the automatic data file generator, the FFT program and 3D visualiser can all be PC-based, whereas, the 3D FDTD modeller could either run on a transputer system, a stand-alone PC, as multi-nodes on a LAN or on a stand-alone high-powered workstation.

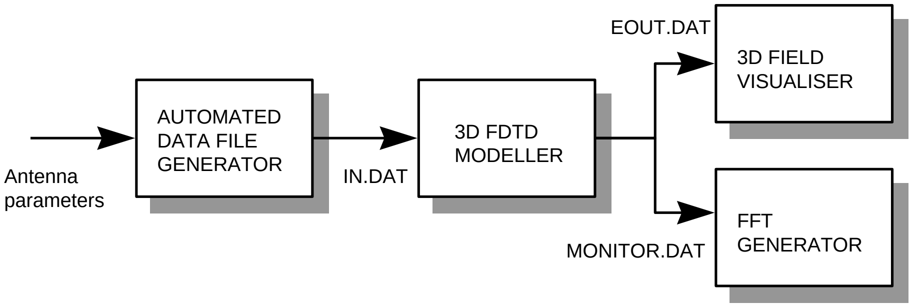


Figure A.1: FDTD method

The 3D FDTD modeller requires an input file (normally, `IN.DAT`). This file defines the number of grid points, the grid size, the location of copper conductors, the dielectric, and so on. This is then read by the 3D FDTD modeller that then outputs to two output data files after each iteration. The first output file stores the electric field surface data for the required slice, normally named `EOUT.DAT`. The second data file stores the electric field at fixed locations in space, this file is normally named `MONITOR.DAT`.

A.2 Microstrip Antenna Modeller

A microstrip antenna modeller was designed and modelled using the equations given in Chapter 4. The input parameters to the program included the resonant frequency and the bandwidth.

A.3 Automatic Data File Generator

After the system has been designed, the data file generator then transposes this data to give physical grid data that the 3D FDTD modeller can use. The output data file is a free format text style with keywords at the start of each line. Table A.1 defines the keywords used in the datafile.

Table A.1: Data file keywords

Keyword	Arguments	Description
NAME	Simulation_name	Simulation name (not used in 3D modeller).
FNAME	File_name	Name of the 3D FDTD modeller output filename (default is EOUT.DAT).
E_R	Dielectric_const	Sets the substrate dielectric constant. If this keyword is not included in the data file then the dielectric constant is assumed to be 2.2.
TSTEP	Time_step	Sets the time step. If this keyword is not include in the data file then the time step is assumed to be given by Equation (3.17).
SLTYPE	Slice_type	Defines where the 2D slice is taken, and is either ‘x’, ‘y’ or ‘z’.
SLICE	Slice	Defines the co-ordinate at which the slice is taken through.
GRID	XN YN ZN TSTEP	Sets up a grid of NX×NY×NZ and simulates the model for TSTEP time-steps.
X	x_1 x_2 x_3 … x_m	Defines the number of points in the *x*-direction and also the size of an element in the x-direction.
Y	y_1 y_2 y_3 … y_n	Defines the number of points in the *y*-direction and also the size of an element in the *y*-direction.
Z	z_1 z_2 z_3 … z_o	Defines the number of points in the *z*-direction and also the size of an element in the *z*-direction.

Table A.1: Data file keywords (continued)

Keyword	Arguments	Description
`;`	comment	Everything after the semi-colon is ignored by the 3D FDTD modeller.
`SOURCE`	x_1 y_1 z_1 x_2 y_2 z_2	Defines a Gaussian pulse source from (x_1,y_1,z_1) to (x_1,y_1,z_1).
`COPPER`	x_1 y_1 z_1 x_2 y_2 z_2	Defines a copper area from (x_1,y_1,z_1) to (x_1,y_1,z_1).
`MONITOR`	x_1 y_1 z_1	Defines a monitor point at (x_1,y_1,z_1). The electric field at this point will be stored in `MONITOR.DAT`.

A.3.1 Microstrip antenna example

Data file A.1 shows an example data file for a microstrip antenna and Figure A.2 shows how the antenna relates to the data file. In this case the grid is 100x100x16 and the model is simulated for 5 000 time steps. The dielectric constant has been set at 2.62. A 2D slice will be taken in the *z*-direction (that is, the *x*-*y* plane) at the Z[1] point. Note that the first point on the *z*-axis is Z[0] and the last point is Z[NZ-1]. A linear grid is used to give each element the dimensions of $1.0\times1.067\times0.597$ mm^3. The total model size is $38\times40.546\times3.582$ mm^3.

The antenna, itself, is defined by two areas of copper, from grid point (18,0,3) to (20,15,3). Thus the *x*-width is two *x*-element, and the length is 15 *y*-elements. The antenna sits on the fourth grid element in the *z*-direction and the head of the antenna is from grid point (5,15,3) to (31,30,3).

The output file for the 2D slice will be stored in `EOUT.DAT` and the default monitor file is taken as `MONITOR.DAT`. The monitor points, in this case, are at (19,1,2), (2,25,3) and (2,25,4), and so on. The first point (19,1,2) is 1 *y*-direction step in from the source.

Data file A.1

```
NAME    Test Run     ; This is just the name of the design
FNAME   EOUT.DAT
GRID    38 38 6 5000 ; XxYxZ (38x38x6) grid points 5000 time steps
SLTYPE  Z        ; valid slices are X/Y or Z (in this case Z-slice)
SLICE   1        ; slice grid point
E_R     2.62
   ; X grid points. Note that the first two grid points
   ; give the dimensions of the grid (e.g. 0.8,0.8,0.5)
X       0 1.0 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
   ; Y grid points
Y       0 1.067 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 33 34 35 36 37 38 39 40 41 42
   ; Z grid points
Z       0 .597 2 3 4 5
   ; Define Source (Xstart Ystart Zstart Xend Yend Zend)
SOURCE  18 0 1 20 0 2 ; this is on the y-normal wall
   ; Define Copper Areas (Xstart Ystart Zstart Xend Yend Zend)
COPPER  18 0 3 20 15 3  ; antenna feed
COPPER  5 15 3 31 30 3  ; antenna feed
MONITOR 19 1 2         ;  monitor source
MONITOR 2 25 3
MONITOR 2 25 4
MONITOR 2 25 5
MONITOR 25 25 4
MONITOR 25 25 5
MONITOR 19 37 4
```

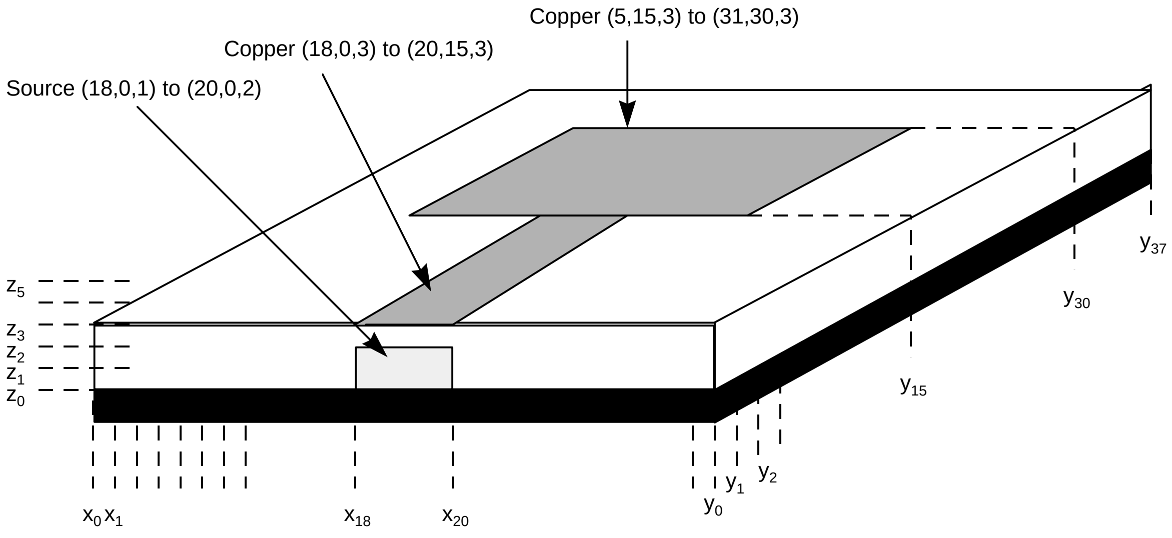


Figure A.2: Microstrip antenna model

A.3.2 PCB example

Data file A.1 shows an example data file for a multi-souce PCB simulation. In this case the grid is a 100x100x16 and the model is to be simulated for 5 000 time steps. The slice taken is a *z*-slice at the fifth grid point (that is, *z*=4). The length of the element size is set by the first two values in the X, Y and Z keywords. In this case the element dimension is 0.389×0.4×0.265 mm^3, which makes the total model size 38.9×40×4.24 mm^3. There are four sources in this example, between (20,0,3) and (25,0,3), (42,0,3) and (46,0,3), (54,0,3) and (58,0,3), and, (76,0,3) and (82,0,3).

Data file A.1: IN.DAT

```
NAME    Test Run   ; This is just the name of the design
FNAME   EOUT.DAT
SLTYPE  Z
SLICE   4
GRID 100 100 16 5000    ; XxYxZ (100x100x16) grid points 5000 time steps
X  0 .389 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73
74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97
98 99
Y  0 .4 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98
99
Z  0 .265 2 3 4 5 6 7 8 9 10 11 12 13 14 15
SOURCE  20 0 3 25 0 3
SOURCE  42 0 3 46 0 3
SOURCE  54 0 3 58 0 3
SOURCE  76 0 3 82 0 3
COPPER  20 0 3 25 60 3
COPPER  0 60 3 25 65 3
COPPER  42 0 3 46 25 3
COPPER  54 0 3 58 25 3
COPPER  42 25 0 58 25 3
COPPER  48 29 3 52 99 3
COPPER  76 0 3 82 50 3
COPPER  70 50 3 88 99 3
MONITOR    22   0  3
MONITOR    44   0  3
MONITOR    56   0  3
MONITOR    80   0  3
MONITOR    22   99 3
MONITOR    50   99 3
```

A.4 3D FDTD Modeller

The source code for the 3D FDTD modeller is available over the Internet from `http://www.eece.napier.ac.uk/res.html`. It has been structured to reduce simulation times, thus there are no time-consuming function calls within the main simulation loop. A truncated example of a surface slice output data file is given in Data file A.2. The start of the file contains a copy of the input data file. This information is used by the surface viewing package to draw the structure. The data for each time step is then inserted after the keyword `DATA`. Next the time step is stored, in the example given the time-steps are at 10, 20 and 30. In most simulations the a *z*-slice is taken through the structure which gives values are in the *x*-*y* plane. The format of the output is then in the form:

E_z[0,0,Z_{SLICE}] E_z[0,1, Z_{SLICE}] E_z[0,2, Z_{SLICE}] … E_z[0,NY-1,ZS Z_{SLICE}]
E_z[1,0, Z_{SLICE}] E_z[1,1, Z_{SLICE}] E_z[1,2, Z_{SLICE}] … E_z[1,NY-1, Z_{SLICE}]
E_z[2,0, Z_{SLICE}] E_z[2,1, Z_{SLICE}] E_z[2,2, Z_{SLICE}] … E_z[2,NY-1, Z_{SLICE}]
.
.
E_z[NX-1,0, Z_{SLICE}] E_z[NX-1,1, Z_{SLICE}] E_z[NX-1,2,Z_{SLICE}]…E_z[NX-1,NY-1, Z_{SLICE}]

Initially all the fields are set to zero, so at the start and end of the simulation there are many zero values. To save storage space a whole line of zero values is stored as $*M*, where *M* is the number of zero values, for example *$38* indentifes *38* zero values. A single zero values is stored as an ‘X’.

In the example data it can be seen that a pulse is begining to propagate out from the source. It can also be seen that most of the values within the model are initially zero.

Chapters 8 and 9 show example of the surfaces produced.

Data file A.2: EOUT.DAT

```
; Run Started 1-0-80 3:58:56
GRID  38 38 6 5000
X  0.00 1.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00 15.00 16.00
17.00 18.00 19.00 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00 30.00
31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00
Y  0.00 1.07 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00 15.00
16.00 17.00 18.00 19.00 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00 28.00 29.00
30.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00 40.00
Z  0.00 0.60 2.00 3.00 4.00 5.00
SOURCE   18 0 1 20 0 2
TSTEP 1.5e-12
E_R   2.62
PULSE 30.00 11.00
COPPER   19 0 3 21 15 3
COPPER   6 15 3 32 30 3
MONITOR  19 1 2
MONITOR  2 25 3
MONITOR  2 25 4
MONITOR  2 25 5
MONITOR  25 25 4
MONITOR  25 25 5
MONITOR  19 37 4
SLICE 1
SLTYPE   Z
DATA
10    ; TIME STEP 10
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
X 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
0.037 0.011 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X
0.037 0.013 0.003 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X
0.037 0.011 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X
X 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
```

```
$38
20          ; TIME STEP 20
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
X 0.003 0.002 0.001 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X
X 0.011 0.007 0.003 0.001 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X
X 0.042 0.022 0.008 0.003 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X
0.438 0.165 0.053 0.017 0.005 0.001 X X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X
0.438 0.190 0.064 0.020 0.006 0.002 X X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X
0.438 0.167 0.055 0.018 0.005 0.001 X X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X
X 0.045 0.024  0.01 0.003 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X
X 0.012 0.008 0.004 0.001 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X
X 0.003 0.002 0.001 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
30             ; TIME STEP 30
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
X X 0.001 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X 0.003 0.003 0.003 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X
X 0.008  0.01 0.007 0.004 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X
```

```
X 0.021 0.025 0.018  0.01 0.005 0.002 X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X
X 0.058 0.059 0.040 0.021 0.009 0.004 0.001 X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X
X 0.160 0.131 0.079 0.040 0.017 0.006 0.002 X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X
1.000 0.462 0.252 0.134 0.064 0.027  0.01 0.003 X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X
1.000 0.529 0.294 0.158 0.076 0.032 0.012 0.004 0.001 X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X
1.000 0.464 0.260 0.144 0.072 0.031 0.012 0.004 0.001 X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X
X 0.178 0.156 0.101 0.053 0.024 0.009 0.003 X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X
X 0.068 0.072 0.051 0.028 0.013 0.005 0.002 X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X
X 0.025 0.030 0.022 0.013 0.006 0.002 X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X
X 0.009 0.011 0.009 0.005 0.002 0.001 X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X
X 0.003 0.004 0.003 0.002 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X
X X 0.001 0.001 X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
$38
```

A.5 FFT Analysis Program

The 3D FDTD modeller outputs a monitor file which can be used to determine the transisient response at monitor points (by default this file is named `MONITOR.DAT`).

Data file A.3 shows a truncated `MONITOR.DAT` file with 15 time-steps. As with `EOUT.DAT` it also contains a copy of the `IN.DAT` file. After the keyword `DATA` the monitor points are listed in columns for each time step. The monitor points can then the processed for a frequency response using an FFT. This could either be achieved using a spread-sheet or with user-writen program.

Data file A.3: MONITOR.DAT

```
; Run Started 1-0-80 3:58:56
GRID 38 38 6 5000
X  0.00 1.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00 15.00
16.00 17.00 18.00 19.00 20.00 21.00 22.00 23.00 24.00 25.00 26.00 27.00
```

```
28.00 29.00 30.00 31.00 32.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00
40.00
Y  0.00 1.07 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00
14.00 15.00 16.00 17.00 18.00 19.00 20.00 21.00 22.00 23.00 24.00 25.00
26.00 27.00 28.00 29.00 30.00 33.00 34.00 35.00 36.00 37.00 38.00 39.00
40.00
Z  0.00 0.60 2.00 3.00 4.00 5.00
SOURCE  18 0 1 20 0 2
TSTEP 1.5e-12
E_R   2.62
PULSE 30.00 11.00
COPPER  19 0 3 21 15 3
COPPER  6 15 3 32 30 3
MONITOR 19 1 2
MONITOR 2 25 3
MONITOR 2 25 4
MONITOR 2 25 5
MONITOR 25 25 4
SLICE 1
SLTYPE  Z
DATA
    1     0.000071     0.000000     0.000000     0.000000     0.000000
    2     0.000228     0.000000     0.000000     0.000000     0.000000
    3     0.000484     0.000000     0.000000     0.000000     0.000000
    4     0.000864     0.000000     0.000000     0.000000     0.000000
    5     0.001424     0.000000     0.000000     0.000000     0.000000
    6     0.002243     0.000000     0.000000     0.000000     0.000000
    7     0.003426     0.000000     0.000000     0.000000     0.000000
    8     0.005094     0.000000     0.000000     0.000000     0.000000
    9     0.007401     0.000000     0.000000     0.000000     0.000000
   10     0.010560     0.000000     0.000000     0.000000     0.000000
   11     0.014852     0.000000     0.000000     0.000000     0.000000
   12     0.020617     0.000000     0.000000     0.000000     0.000000
   13     0.028228     0.000000     0.000000     0.000000     0.000000
   14     0.038059     0.000000     0.000000     0.000000     0.000000
   15     0.050470     0.000000     0.000000     0.000000     0.000000
```

A.6 3D EM Field Visualiser

The 3D EM field visualiser reads the output file from the 3D FDTD modeller (by default this is `EOUT.DAT`). It uses hiden line removal to view pulse propagation in steps of time. The methods used are discussed in more detail in Appendix B.

APPENDIX B

Field Visualisation

B.1 Introduction

The 3D EM field visualiser reads the output file from the 3D FDTD modeller (by default this is EOUT.DAT). It uses hidden line removal to view pulse propagation in steps of time. A bi-cubic B-spline approximation for the surface is used with forward difference polynomial evaluation and other optimising methods to create smooth graphics and with high-speed animation.

A surface plots can be formed by simply joining the 3D data points. This will give a series of rectangles with abrupt changes where adjacent points differ greatly. To obtain a smooth surface plot it is necessary to use polynomial functions to model the surface. This will replace the rectangles with curved surface patches. The polynomial surface function also allows for the calculation of the field amplitude at any point on the surface.

In the most extreme case it is possible to form unique polynomials for two parametrics based on all the points on the surface. Thus for a surface which is n by n points this results in a polynomial of the order of (n-1) and will exactly describe the surface. It would, however, be computationally expensive and suffer from the characteristic instability of high order polynomials.

A better solution is to use a piece-wise polynomial function. This limits the number of points taken into consideration at any one time. Cubic polynomials are the lowest order functions that give the required continuity of a smooth curve or surface.

B.2 Cubic B-splines

Cubic splines [B.1] provide a piece-wise polynomial which describe the entire surface. They have a constant curvature to gives a smooth curve or surface. Uniform cubic B-splines have a uniform knot sequence. This application uses a knot sequence composed of successive integers. A unit difference of successive knots simplifies the algebra to form the segment evaluation expressions. The bi-cubic surface function is a direct extension of the cubic curve function.

Uniform cubic B-splines are used to form bi-cubic expressions in two parametrics descriptive of a surface patch in the centre of a square of 16 control points. Figure B.1 show the relationship of the 16 control points to the surface patch. By varying the two parametrics u and v between 0 and 1, it is possible to calculate any point on the surface of the patch. A different bi-cubic expression is formed for each successive surface patch. The uniform bi-cubic B-spline surface basis equation is given by:

$$\begin{aligned} P(u,v) = \quad & C_{03}v^3 + C_{02}v^2 + C_{01}v + C_{00} + C_{13}v^3u + C_{12}v^2u + C_{11}vu + C_{10}u + \\ & C_{23}v^3u^2 + C_{22}v^2u^2 + C_{21}vu^2 + C_{20}u^2 + \end{aligned}$$

$$C_{33}v^3 u^3 + C_{32}v^2u^3 + C_{31}vu^3 + C_{30} u^3 \qquad (B.1)$$

where,

$C_{00}=[(d_{i-1,j-1}+d_{i-1,j+1}+d_{i+1,j-1}+d_{i+1,j+1})+$

$\quad 4\times(d_{i-1,j}+d_{i,j-1}+d_{i,j+1}+d_{i+1,j})+16\times d_{i,j}]/36;$

$C_{01}=[(d_{i-1,j+1}-d_{i-1,j-1}-d_{i+1,j-1}+d_{i+1,j+1})+4\times(d_{i,j+1}-d_{i,j-1})]/12;$

:::::: ::::::::::

$C_{33}=[(d_{i-1,j-1}-d_{i-1,j+2}+d_{i+2,j-1}+d_{i+2,j+2})+$

$\quad 3\times(-d_{i-1,j+2}+d_{i-1,j+1}-d_{i,j-1}+d_{i,j+2}+$

$\quad d_{i+1,j-1}-d_{i+1,j+2}+d_{i+2,j}-d_{i+2,j+1}) +$

$\quad 9\times(d_{i,j}-d_{i,j+1}-d_{i+1,j}+d_{i+1,j+1})]/36;$

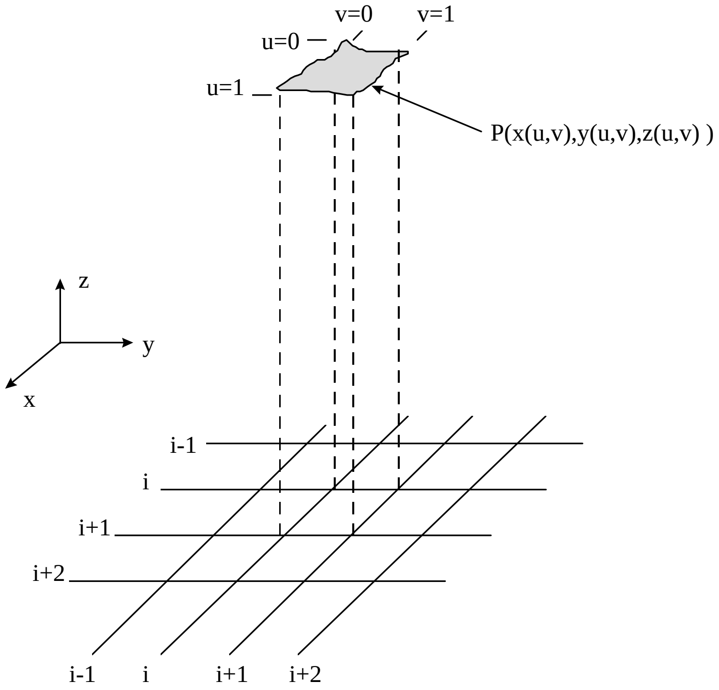


Figure B.1: Surface patch

The bi-cubic function is only evaluated at the edge of the patch. Thus the bi-cubic z function is calculated at the edges, this occurs when u or v is either a 0 or a 1.

For the back segment (u=0):

$$Z(v)= C_{03}v^3+ C_{02}v^2 + C_{01}v + C_{00} \qquad (B.2)$$

for the front segment (u=1):

$$\begin{aligned} Z(v) = \ & (C_{03}+ C_{13} + C_{23}+ C_{23})\, v^3 + \\ & (C_{02}+ C_{12} + C_{22}+ C_{32})\, v^2+ \\ & (C_{01}+ C_{11} + C_{21}+ C_{31})\, v + \\ & (C_{00}+ C_{10} + C_{20}+ C_{30}) \qquad (B.3) \end{aligned}$$

for the left segment (v=0):

$$Z(u)= C_{30}u^3+ C_{20}u^2 + C_{10}u + C_{00} \qquad (B.4)$$

for the right segment (v=1):

$$\begin{aligned} Z(u) = \ & (C_{33}+ C_{32} + C_{31}+ C_{30})\, u^3 + \\ & (C_{23}+ C_{22} + C_{21}+ C_{20})\, u^2+ \\ & (C_{13}+ C_{12} + C_{11}+ C_{10})\, u + \\ & (C_{03}+ C_{02} + C_{01}+ C_{00}) \qquad (B.5) \end{aligned}$$

Equation (B.2)-(C5) can be used to draw each of the sides of the patches.

B.3 Other methods

A cubic forward difference polynomial evaluation [B.2] can be used to evaluate segment evaluation cubics (B.2)-(B.5). This further simplies the surface rendering and increases the speed of animation. The forward difference method eliminates the need for floating-point multiplication when evaluating the polynomials. Once the initial values of the function and difference terms have been established, it allows evaluation of polynomials at discrete intervals using integer addition.

B.4 Viewing operations

A 4×4 matrix performs 3D translation of points, centering the surface over the 3D origin. Scaling of 3D points is carried out by a 4×4 matrix, creating a 3D surface image.

B.4.1 3D translation

The 3D transulation convers the points in the x- and y-direction so that the surface is centered around the origin.

$$\begin{bmatrix} 1 & 0 & 0 & TransX \\ 0 & 1 & 0 & TransY \\ 0 & 0 & 1 & TransZ \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.6}$$

where

$TransX = -(\text{X grid points} - 1)/2,\ TransY = -(\text{Y grid points} - 1)/2$, TransZ=0.

B.4.2 3D scaling

The 3D scaling scales the surface in 3-dimensions. The *ScaleX* and *ScaleY* values are defined to allow the largest surface displable on the screen at a 45° rotation.

$$\begin{bmatrix} ScaleX & 0 & 0 & 0 \\ 0 & ScaleY & 0 & 0 \\ 0 & 0 & ScaleZ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.7}$$

where

ScaleX = ResX / ((X grid points)^2 + (Y grid points) ^{2})

ScaleY = ResY / ((X grid points)^2 + (Y grid points) ^{2})

ScaleZ=150 (by default)

B.4.3 3D rotation

The 3D rotation rotates the 3D points about the x, y and x axis by the angles q_X, q_Y, and q_Z.

$$\begin{bmatrix} Cz.Cy & Cz.Sy.Sx - Cx.Sz & Cz.Cx.Sy + Sz.Sx & 0 \\ Cy.Sz & Cz.Cx + Sz.Sy.Sx & Cx.Sz.Sy - Cz.Sx & 0 \\ -Sy & Cy.Sx & Cy.Cx & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{B.8}$$

where C_X = cos(q_X), C_Y= cos(q_Y), C_Z = cos(q_Z), S_X = sin(q_X), S_Y= sin(q_Y), S_Z = sin(q_Z).

B.4.4 Projection

If the electric field point E is defined as (x1,y2,z2, EFIELD) and it is projected into the view plane equation defined by m (m0,m1,m2,m3), then:

$$\mathrm{P=E.m^T – (E.m)\ I_4} \tag{B.9}$$

is a 4×4 matrix that, when matrix-mulitplied by 3 dimensional homogeneous coordinate points, projects 3D points onto a plane, m. I_4 defines the 4×4 identity matrix

B.4.5 Transformation from 3D to 2D co-ordinates

The graphics display has a 2D co-ordinate system, thus a 3D-to-2D transformation is required. If A (a,b,c,d) defines the vector equation of the view plane for the x-axis, B (e,f,g,h) defines the vector equation of the view plane for the y-axis and C (i,j,k,l) defines the vector equation of the view plane for the z-axis then the transformation matrix will be:

$$N=(H^T.H)^{-1}.H^T \tag{B.10}$$

where

$$H=\begin{bmatrix} a & e & i \\ b & f & j \\ c & g & k \\ d & h & l \end{bmatrix} \tag{B.11}$$

When the N matrix is mulitplied by a 3D point in the view plane then a 2D projection results.

B.4.6 2D translation and scaling to screen co-ordinates

Finally the 2D translation can be mapped to the screen co-ordinates. If the screen has co-ordinates of ($MinX$, $MinY$) to ($MaxX$,$MaxY$) and the current view plane is ($ViewPlaneMaxX$,$ViewPlaneMaxY$) then 3x3 translation matrix is:

$$\begin{bmatrix} ScreenScaleX & 0 & ScreenTransX \\ 0 & ScreenScaleY & ScreenTransY \\ 0 & 0 & 1 \end{bmatrix} \quad (B.12)$$

where:

$ScreenTransX = (MaxX + MinX) / 2;$

$ScreenTransY = (MaxY + MinY) / 2;$

$ScreenScaleX = (MaxX + MinX) / (2\text{x}ViewPlaneMaxX);$

$ScreenScaleY = (MaxY + MinY) / (2\text{x}ViewPlaneMaxY);$

B.5 Bresenham's line algorithm

The Bresenham's line drawing algorithm [B.3] has been used in the surface drawing as it has been optimised to take advantage of the video graphics used in the PC.

B.6 Graphics implementation

The source code for the graphics viewer can be down-loaded from the Internet page:

```
http: www.eece.napier.ac.uk/~bill_b/res.html
```

B.7 References

[B.1] Bartels RH, Beatty JC and Barsky BA, "An Introduction to Splines for Use in Computer Graphics and Geometric Modelling", Morgan Kaufman Publishers, Los Altos California.

[B.2] Hearn D and Baker M, "Computer Graphics", Prentice Hall, 1986, pp 204-

205.

[B.3] Hearn D and Baker M, “Computer Graphics”, Prentice Hall, 1986, pp 58-61.

APPENDIX C

Gaussian Pulse Analysis

C.1 Introduction

The 3D FDTD method can be used any type of input signal it normally uses either a Gaussian pulse or a sine-wave. A Gaussian pulse has the advantage over sine-wave in that it contains a wide band of frequencies. The maximum significant frequency within the pulse can simply be set by adjusting its width.

C.2 Frequency response

The standard form of a Gaussian pulse is:

$$f(t) = e^{-at^2} \tag{C.1}$$

The fourier transform of this is then:

$$F(j\omega) = \int_{-\infty}^{\infty} e^{-at^2} e^{-j\omega t^2}\, dt \tag{C.2}$$

completing the square of the exponential by multiplying and dividing by $e^{\frac{\omega^2}{4a}}$ gives:

$$F(j\omega) = e^{-\frac{\omega^2}{4a}} \int_{-\infty}^{\infty} e^{-\left(\sqrt{a}t + j\frac{\omega}{2\sqrt{a}}\right)^2} dt \tag{C.3}$$

Changing the integration variable to:

$$x = \sqrt{a}t + j\frac{\omega}{2\sqrt{a}} \tag{C.4}$$

then $dx = \sqrt{a}dt$ (C.5)

This gives:

$$F(j\omega) = \frac{e^{-\frac{\omega^2}{4a}}}{\sqrt{a}} \int_{-\infty}^{\infty} e^{-x^2}\, dx \tag{C.6}$$

$$F(j\omega) = \sqrt{\frac{\pi}{a}}\, e^{-\frac{\omega^2}{4a}} \qquad \text{because } \int_{-\infty}^{\infty} e^{-x^2}\, dx = \sqrt{\pi} \tag{C.7}$$

Thus $f(t) = e^{-at^2}$ transforms to $F(j\omega) = \sqrt{\frac{\pi}{a}}\, e^{-\frac{\omega^2}{4a}}$

APPENDIX D

Microstrip Design Examples

D.1 Introduction

Microstrip is used to guide electromagnetic waves around microwave circuits. It is simple to manufacture and has been modelled by many researchers over the years. This appendix discusses the calculations used in the design of microstrip lines and microstrip antennas.

D.2 Microstrip design

Figure D.1 shows a cross section of a microstrip line. The characteristic impedance of the line varies as the width of the line and with the thickness of the lines [D.1]. Equations (D.1)-(D.7) can be used to determine its characteristic impedance and the equivalent direct constant (e_{eff}) .

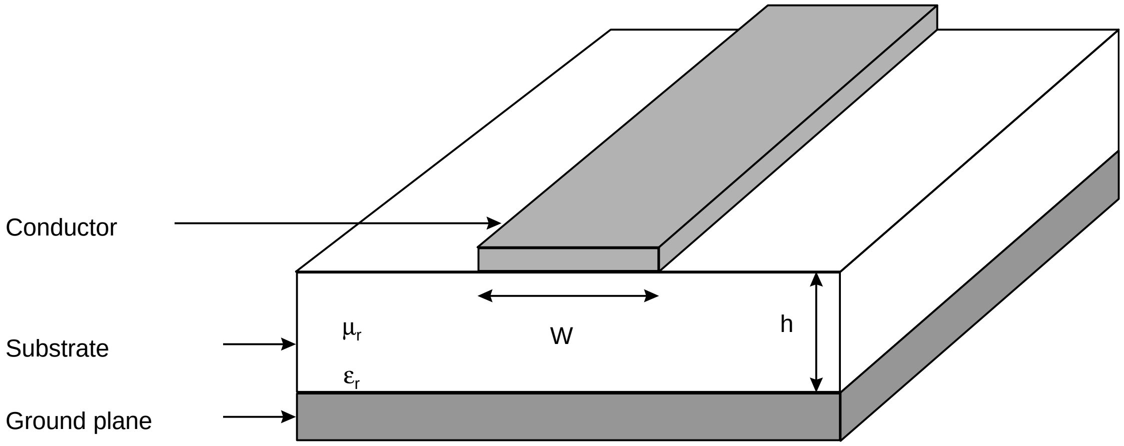


Figure D.1: Microstrip track

$$A = \frac{120}{\sqrt{2(\varepsilon_r + 1)}} \tag{D.1}$$

$$B = \frac{1}{2}\left[\frac{\varepsilon_r - 1}{\varepsilon_r + 1}\right].\left[ln\frac{\pi}{2} + \frac{1}{\varepsilon_r} ln\frac{4}{\pi}\right] \tag{D.2}$$

$$C = ln\frac{8h}{W} + \frac{1}{32}\left[\frac{W}{h}\right]^2 \tag{D.3}$$

$$D = \frac{60\pi}{\sqrt{\varepsilon_r}} \tag{D.4}$$

$$E = \frac{W}{2h} + 0.4413 + 0.08226\left[\frac{\varepsilon_r - 1}{\varepsilon_r{}^2}\right] + \frac{\varepsilon_r + 1}{2\pi\varepsilon_r}\left[1.542 + ln\left(\frac{W}{2h} + 0.94\right)\right] \tag{D.5}$$

If $\frac{W}{h}$ is less than 1.3 then $\varepsilon_{eff} = \dfrac{\varepsilon_r + 1}{2\left(1 - \frac{B}{C}\right)^2}$ (D.6)

else $\varepsilon_{eff} = \dfrac{\varepsilon_r + 1}{2} + \left(\dfrac{\varepsilon_r - 1}{2}\right)\left(1 + \dfrac{10h}{W}\right)^{-\frac{1}{2}}$ (D.7)

If $\frac{W}{h}$ is less than 3.3 then $Z_0 = A(C - B)$ (D.8)

else $Z_0 = \frac{D}{E}$ (D.9)

D.3 Microstrip design example

Most high-frequency systems are matched to 50 Ω. Thus a good design example is to determine the track width for a matched line. The design parameters are:

W= 2.46 mm, *h*=0.794 mm and e_r=2.2.

Using (D.3), (D.4) and (D.9), gives:

$$D = \frac{60\pi}{\sqrt{\varepsilon_r}} = \frac{60\pi}{\sqrt{2.2}} = 127.0836$$

$$E = \frac{W}{2h} + 0.4413 + 0.08226\left[\frac{\varepsilon_r - 1}{\varepsilon_r{}^2}\right] + \frac{\varepsilon_r + 1}{2\pi\varepsilon_r}\left[1.542 + ln\left(\frac{W}{2h} + 0.94\right)\right]$$

$$E = \frac{2.46}{2 \times 0.794} + 0.4413 + 0.08226\left[\frac{2.2 - 1}{2.2^2}\right] + \frac{2.2 + 1}{2\pi 2.2}\left[1.542 + ln\left(\frac{2.46}{2 \times 0.794} + 0.94\right)\right]$$
$$= 2.5581$$

$$Z_0 = \frac{127.0836}{2.5581} = 49.69\ \Omega$$

It can also be shown from [D.7] that e_{eff} is 1.8918.

D.4 Microstrip antenna design

It was shown in Chapter 4 that the width and length of a microstrip patch antenna can be found by a mixture of analytical analysis and empirical methods.

D.4.1 Antenna width

The width of the antenna can be found from:

$$W = \frac{c}{2f_r}\left(\frac{e_r + 1}{2}\right)^{-\frac{1}{2}} \tag{D.10}$$

As stated in Chapter 4, the *c* divided by 2f_r term gives one-half a wavelength in free-space and the second term scales it to give a half-wavelength in the substrate.

D.4.2 Antenna length

The length is also found by calculating the half-wavelength value and then subtracting a small length to take into account the fringing fields, it is given by:

$$L = \frac{c}{2f_r\sqrt{e_e}} - 2.\Delta l \tag{D.11}$$

where

$$\Delta l = 0.412h\frac{(e_e + 0.3)\cdot\left(\frac{W}{h} + 0.264\right)}{(e_e - 0.258)\cdot\left(\frac{W}{h} + 0.8\right)} \tag{D.12}$$

and

$$e_e = \frac{e_r + 1}{2} + \frac{e_r - 1}{2}\left[1 + \frac{12t}{W}\right]^{-\frac{1}{2}} \tag{D.13}$$

D.4.3 Input admittance

The microstrip antenna must be properly matched to the input supply in order to mimimise reflections and maximise power transfer. Thus the input impedance or admittance of the microstrip antennas must be matched to the line feed, normally 50 Ω. The two main methods used are to match the antenna to the source using a quarter-wave transformer or by offseting the line feed by a designated offset.

The discontinuity between the line feed and the antenna head can be modelled by a shunt conductance *G* and a shunt capacitance *jB*, as shown in Figure D.2. Richards *et al.* [D.2], Bhal [D.3], Carver [D.4] have derived an equivalent model and formula for input impedance at a distance offset by *z* is:

$$Y(z) = 2G\left[cos^2(\beta z) + \frac{G^2 + B^2}{{Y_0}^2}sin^2(\beta z) + \frac{B}{Y_0}sin(2\beta z)\right]^{-1} \tag{D.14}$$

where

$$G = \frac{1}{R_r} \tag{D.15}$$

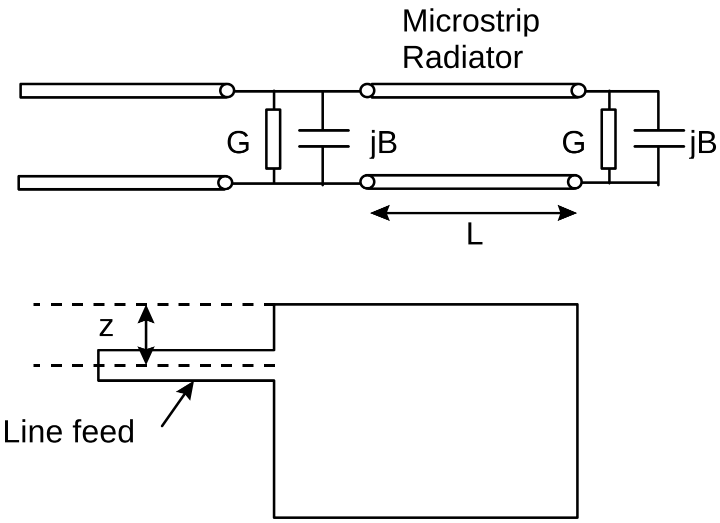


Figure D.2: Equivalent circuit for a microstrip antenna

$$R_r = \frac{120\lambda_0}{1 - \frac{(k_0 h)}{24}} \tag{D.16}$$

$$B = \frac{k_0 \Delta l \sqrt{\varepsilon_e}}{Z_0} \tag{D.17}$$

$$\beta = \frac{2\pi\sqrt{\varepsilon_e}}{\lambda_0} \tag{D.18}$$

D.5 Microstrip antenna example

The microstrip antenna simulated in Chapter 8 is designed to resonate at 7.5 GHz. Its width is 12.45 mm and its length is 16 mm. The substrate used has a dielectric constant of 2.2 and has a thickness of 0.794 mm. The feed width is 2.46 mm, which is designed to give a characteristic impedance of 50 Ω, see calculation in section D.3.

The antenna feed is then offset by a distance *z* to match the antenna to the line feed. Using D.14 gives:

$$\lambda_0 = \frac{c}{f} = \frac{3 \times 10^8}{7.5 \times 10^9} = 0.4 \text{ m}$$

$$k = \frac{2\pi f}{c} = \frac{2 \times \pi \times 7.5 \times 10^8}{3 \times 10^8} = 157.08$$

$$R_r = \frac{120\lambda_0}{1 - \frac{(k_0 h)}{24}} = \frac{120 \times 0.4}{1 - \frac{(157.08 \times 0.794 \times 10^{-3})}{24}} = 4.8$$

$$\varepsilon_e = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[1 + \frac{12t}{W}\right]^{-\frac{1}{2}} = 2.07$$

It can be shown from (D.9) that Z_0 is 13.9 Ω (1 / Y_0), thus:

$$B = \frac{k_0 \Delta l \sqrt{\varepsilon_e}}{Z_0} = 0.012 \quad \text{and} \quad G = \frac{1}{R_r} = \frac{1}{4.8} = 0.208$$

The equation for the admittance (given below) can then be calculated for steps of z until a match is found, that is, when Y(z) is 0.02 S (or 50 Ω).

$$Y(z) = 2G\left[cos^2(\beta z) + \frac{G^2 + B^2}{{Y_0}^2} sin^2(\beta z) + \frac{B}{Y_0} sin(2\beta z)\right]^{-1}$$

Table D.1 shows a sample run with varying offsets. It can be seen from the table that the input line should be offset by 3.27 mm to produce an input impedance of 50 Ω. The final design is shown in Figure D.3.

Table D.1: Input impedance for varying input line offsets

z (mm)	*Zin(z)*	*z*	*Zin(z)*
0	2.4	3.05	31.1
0.5	2.7	3.10	36.7
1.0	3.7	3.15	39.99
1.5	5.5	3.20	43.78
2.0	8.9	3.25	48.1
2.5	15.6	3.3	53.07
3.0	31.2	3.5	85.6

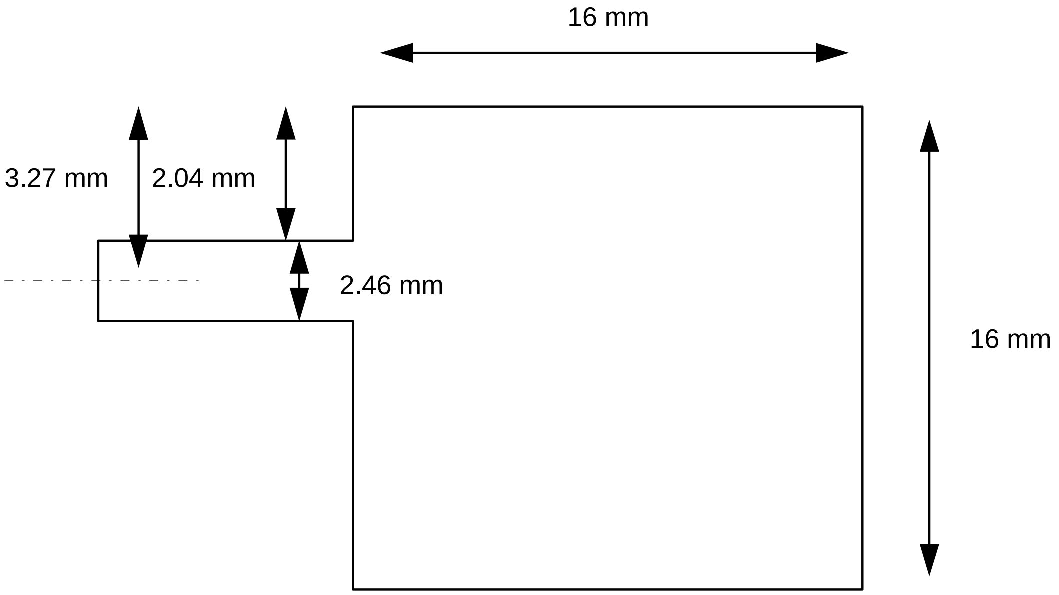


Figure D.3: Microstrip antenna design

D.6 References

[D.1] Fusco V, “Microwave Circuits: Analysis and Computer-Aided Design”, Prentice-Hall International, 1987.

[D.2] Richards, WF, Lo YT and Harrison DD, “Theory and Experiment on Microstrip Antennas”, *Electronic Letters*, Vol. 15, 1979, pp 42-44.

[D.3] Bahl IJ, Bhartia P, "Microstrip Antennas", *Artech House*, 1980.

[D.4] Carver KR, "Practical Analytical Techniques for the Microstrip Antenna", *Proc. Wordshop on Printed Circuit Antennas*, Oct 1979, pp 7.1- 7.20.

[D.5] James JR & Hall PS, "Handbook of Microstrip Antennas", *IEE Electromagnetic Waves Series*, No. 28, Peter Peregrinus, 1989. Vol. 1.

BIBLIOGRAPHY

[1.1] Buchanan WJ, Gupta NK, “Simulation of Three-Dimensional Finite-Difference Time-Domain Method on Limited Memory Systems”, *International Conference on Computation in Electromagnetics*, IEE, Savoy Place, London, 1991.

[1.2] Buchanan WJ, Gupta NK, “Parallel Processing of the Three-Dimensional Finite-Difference Time-Domain Method”, *National Radio Science Colloquium*, University of Bradford, 7-8 Jul. 1992.

[1.3] Buchanan WJ, Gupta NK, “Computers as Applied to Time-Domain Methods in Electrical Engineering”, *World Conference on Engineering Education*, University of Portsmouth, 20-25 Sep. 1992.

[1.4] Buchanan WJ, Gupta NK, “Simulation of Near-Field Radiation for a Microstrip Antenna using the 3D-FDTD Method”, *NRSC ‘93*, University of Leeds, Apr. 1993.

[1.5] Buchanan WJ, Gupta NK and Arnold JM, “Simulation of Radiation from a Microstrip Antenna using Three-Dimensional Finite-Difference Time-Domain (FDTD) Method”, *IEE’s Eighth International Conference on Antennas and Propagation*, Heriot-Watt University, Edinburgh, Apr. 1993.

[1.6] Buchanan WJ, Gupta NK, “Simulation of Electromagnetic Pulse Propagation in Three-Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques”, *Electrosoft ‘93*, Jul. 1993, Southampton.

[1.7] Buchanan WJ, Gupta NK, “Maxwell’s Equations in the 21st Century”, *IJEEE*, vol. 30, 4, Oct. 1993.

[1.8] Buchanan WJ, Gupta NK, “Parallel Processing Techniques in EMP Propagation using 3D Finite-Difference Time-Domain (FDTD) Method”, *Journal of Advances in Engineering Software*, vol. 18, 3, 1993.

[1.9] Buchanan WJ, Gupta NK and Arnold JM, “3D FDTD Method in a Microstrip Antenna’s Near-Field Simulation”, *Second International Conference on Computation in Electromagnetics*, Apr. 1994.

[1.10] Buchanan WJ, Gupta NK and Arnold JM, “Application of 3D Finite-Difference Time-Domain (FDTD) Method to Predict Radiation from a PCB with High Speed Pulse Propagation”, *Ninth International Conference on Electromagnetic Compatibility*, University of Manchester, UK, Aug. 1994.

[1.11] Buchanan WJ, Gupta NK, “An Accurate Model for the Parallel Processing of the 3D Finite-Difference Time Domain (FDTD) Method in the Simulation of Antenna Radiation”, *QMW 1996 Antenna Symposium*, Jul. 1995.

[1.12] Buchanan WJ, Gupta NK, “Prediction of Electric Fields from Conductors on a PCB by 3D Finite-Difference Time-Domain Method”, *IEE’s Engineering, Science and Education Journal*, Aug. 1995.

[1.13] Buchanan WJ, Gupta NK, “Prediction of Electric Fields in and around PCBs – 3D Finite-Difference Time-Domain Approach with Parallel Processing”, *Journal of Advances in Engineering Software*, Dec. 1995.

[1.14] Buchanan WJ, Gupta NK, “Machine Independent Algorithm for Concurrent Finite-Element Problems”, *IEE CEM ‘96*, University of Bath, Apr. 1996.

[1.15] Buchanan WJ, Gupta NK, “Solving Finite-Element Problems using the Parallel Gaussian Elimination Technique”, *Electrosoft ‘96*, Jun. 1996.

[16] D Sheen, S Ali, M Abouzahra, and J Kong, “Application of Three-Dimensional Finite-Difference Method to the Analysis of Planar Microstrip Circuits”, IEEE MTT, Vol. 38, pp 849-857, July 1990.

[17] X Zang, J Fang and K Mei, “Calculations of the dispersive characteristics of microstrips by the FDTD method”, IEEE MTT, Vol. 26, pp 263-267, Feb 1988.

[18] V Svetlana, K Yee and K Mei, “A Subgridding Method for the Time-Domain Finite-Difference Method to Solve Maxwell’s Equations”, IEEE MTT, Vol. 39, No 3, March 1991.

[19] Railton C and McGeehan, “Analysis of microstrip discontinuities using the FDTD method”, MWSYM 1989, pp1089-1012.

[20] Shibata T, Havashi T and Kimura T, “Analysis of microstrip circuits using three-dimensional full-wave electromagnetic field analysis in the time-domain”, IEEE MTT, Vol. 36, pp1064-1070, June 1988.

[21] A Taflove, “The Finite-Difference Time-Domain Method for Electromagnetic Scattering and Interaction Problems”, IEEE Trans. Electromagnetic Compatibility, Vol. EMC-22, pp191-202, Aug 1980.

[22] Hese J and Zutter D, “Modelling of Discontinuities in General Coaxial Waveguide Structures by the FDTD-Method”, IEEE MTT, Vol. 40, March 1992.

[23] Railton CJ, Richardson KM, McGeehan JP and Elder KF, “The Prediction of Radiation Levels from Printed Circuit Boards by means of the FDTD Method”, IEE International Conference on Computation in Electromagnetics, Savoy Place, London, November 1991.

[24] WJ Buchanan, NK Gupta, “Simulation of three-dimensional finite-difference time-domain method on limited memory systems”, *International Conference on Computation in Electromagnetics*, IEE, Savoy Place, London, 1991.

[25] WJ Buchanan, NK Gupta “Parallel Processing of the Three-Dimensional Finite-Difference Time-Domain Method”, *National Radio Science Colloquium*, University of Bradford, 7-8 July 1992.

[26] WJ Buchanan, NK Gupta “Computers as Applied to Time-Domain Methods in Electrical Engineering”, *World Conference on Engineering Education*, University of Portsmouth, 20-25 September 1992.

[27] WJ Buchanan, NK Gupta and JM Arnold, “Simulation of Radiation from a Microstrip Antenna using Three-Dimensional Finite-Difference Time-Domain (FDTD) Method”, *IEE Eight International Conference on Antennas and Propagation*, Heriot-Watt University, Edinburgh, April 1993.

[28] WJ Buchanan, NK Gupta, “Simulation of Electromagnetic Pulse Propagation in Three-Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques”, *Electrosoft 93*, July 1993, Southampton.

[29] WJ Buchanan, NK Gupta, “Maxwell’s Equations in the 21st Century, *IJEEE*, Vol. 30, No. 4, October 1993.

[30] WJ Buchanan, NK Gupta, “Simulation of near-field radiation for a microstrip antenna using the 3D FDTD method”, *NRSC 93*, University of Leeds, April 1993.

[31] WJ Buchanan, NK Gupta and JM Arnold, “3D FDTD Method in a Microstrip Antenna’s Near-Field Simulation”, *Second International Conference on Computation in Electromagnetics*, April 1994.

[32] James J.R, Hall PS and Wood C,”Microstrip antenna theory and design”,Peter Peregrinus, 1981.

[33] G. Mur “Absorbing boundary condictions for the finite difference approximation of the time domain electromagnetic field equations”,*IEEE Trans. Electromagnetic Compat*.,vol EMC-23,pp 377-382, Nov 1981.

[34] B Enquist and A Majda,”Absorbing boundary conditions for the numerical simulation of waves”,*Math. Computations*, Vol. 31, No. 138, pp 629-651, July 1977.

[35] WJ Buchanan, NK Gupta, “Parallel Processing Techniques in EMP Propagation using 3D Finite-Difference Time-Domain (FDTD) Method”, *Journal of Advances in Engineering Software*, Vol 18, No. 3, 1993.

[36] WJ Buchanan, NK Gupta,"Prediction of Electric Fields from Conductors on a PCB by 3D Finite-Difference Time-Domain Method", *IEE Engineering, Science and Education Journal* (accepted December 1994).

[37] WJ Buchanan, NK Gupta, "Simulation of Electromagnetic Pulse Propagation in Three-Dimensional Finite Difference Time-Domain Method using Parallel Processing Techniques", *Electrosoft 93*, July 1993, Southampton, UK.

[38] WJ Buchanan, NK Gupta and JM Arnold, "3D FDTD Method in a Microstrip Antenna's Near-Field Simulation", *Second International Conference on Computation in Electromagnetics*, April 1994.

[39] WJ Buchanan, NK Gupta and JM Arnold, "Application of 3D Finite-Difference Time-Domain (FDTD) method to predict radiation from a PCB with high speed pulse propagation", *Ninth International Conference on Electromagnetic Compatibility*, University of Manchester, UK, August 1994.